Bansilal Ramnath Agarwal Charitable Trust’s
Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to University of Pune)

Structure & Syllabus of
B.Tech. (Information Technology)

Pattern ‘F-11’
Effective from Academic Year 2014-15

Prepared by: - Board of Studies in Information Technology

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Signed by

Chairman – BOS    Chairman – Academic Board

Page 1 of 251
## Contents

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Program Educational Objectives &amp; Program Outcomes of B.Tech</strong> (Information Technology)</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td><strong>Course Structure - Module I and II</strong></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>2.1 CS10102 Computer programming (Theory Course)</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>2.2 CS10302 Computer Programming (Laboratory)</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td><strong>Course Structure - Module III</strong></td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td><strong>Course Syllabi for Courses - Module III</strong></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>4.1 CS20117 Discrete Structures and Graph Theory (Theory Course)</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>4.2 CS20111 Data Structures (Theory Course)</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>4.3 CS20108 Computer Organization (Theory Course)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>4.4 CS21109 Applied Electronics (Theory Course)</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>4.5 CS20113 Digital Electronics and Logic Design (Theory Course)</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>4.6 CS20115 Discrete Structures (Tutorial)</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>4.7 CS20113 Digital Electronics and Logic Design (Tutorial)</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>4.8 CS20311 Data structures (Laboratory Course)</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>4.9 CS20313 Digital Electronics (Laboratory Course)</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>4.10 CS27401 Mini Project</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>4.11 CS24301 VB.NET OR</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>CS24303 C#.NET (Skill Development Course)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>4.12 Elective –Soft Skills</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.13 CS20401 Comprehensive Viva Voce</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.14 Institute Elective</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Course Structure - Module IV</strong></td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td><strong>Course Syllabi for Courses - Module IV</strong></td>
<td>51</td>
</tr>
</tbody>
</table>


Page 2 of 251
| 6.1  | CS21104 | Mathematical Transformations and Applications (Theory Course) | 53 |
| 6.2  | CS20110 | Computer Graphics (Theory Course) | 56 |
| 6.3  | CS20105 | Principles of Programming Languages (Theory Course) | 59 |
| 6.4  | CS21112 | Data Communication (Theory Course) | 61 |
| 6.5  | CS20114 | Microprocessor and Interfacing (Theory Course) | 65 |
| 6.6  | CS20114 | Microprocessor and Interfacing (Tutorial) | 67 |
| 6.7  | CS21104 | Mathematical Transforms and Applications (Tutorial) | 70 |
| 6.8  | CS20314 | Microprocessor and Interfacing (Laboratory Course) | 71 |
| 6.9  | CS20305 | Principles of Programming Languages (Laboratory Course) | 74 |
| 6.10 | CS20310 | Computer Graphics (Laboratory Course) | 76 |
| 6.11 | CS27402 | Mini Project | 77 |
| 6.12 | CS24302 | ASP.NET OR | 78 |
| 6.13 | CS24304 | Python (Skill Development Course) | 80 |
| 6.14 | CS20402 | Comprehensive Viva Voce | - |

### Course Structure - Module V

| 7    | Course Syllabi for Courses - Module V | 84 |

| 8.1  | CS30101 | Operating Systems (Theory Course) | 87 |
| 8.2  | CS30116 | Computer Networks (Theory Course) | 90 |
| 8.3  | CS30105 | Theory of Computation (Theory Course) | 93 |
| 8.4  | CS30117 | Human Computer Interaction (Theory Course) | 96 |
| 8.5  | CS30109 | Web Technologies (Theory Course) | 99 |
| 8.6  | CS30101 | Operating Systems (Tutorial) | 102 |
| 8.7  | CS30117 | Human Computer Interaction (Tutorial) | 104 |
| 8.8  | CS30303 | Operating Systems (Laboratory Course) | 105 |
| 8.9  | CS30316 | Computer Networks (Laboratory Course) | 107 |
| 8.10 | CS30309 | Web Technologies (Laboratory Course) | 109 |
## Course Structure - Module VI

### Course Syllabi for Courses - Module VI

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS30102</td>
<td>Software Engineering (Theory Course)</td>
<td>118</td>
</tr>
<tr>
<td>CS30106</td>
<td>Database Management Systems (Theory Course)</td>
<td>121</td>
</tr>
<tr>
<td>CS30118</td>
<td>Advanced Data Structures and Algorithms (Theory Course)</td>
<td>124</td>
</tr>
<tr>
<td>CS30114</td>
<td>Systems Programming (Theory Course)</td>
<td>127</td>
</tr>
<tr>
<td>CS31119</td>
<td>Object Oriented Modeling and Design (Theory Course)</td>
<td>130</td>
</tr>
<tr>
<td>CS30102</td>
<td>Software Engineering (Tutorial)</td>
<td>133</td>
</tr>
<tr>
<td>CS30118</td>
<td>Advanced Data Structures and Algorithms (Tutorial)</td>
<td>135</td>
</tr>
<tr>
<td>CS30314</td>
<td>System programming (Laboratory Course)</td>
<td>137</td>
</tr>
<tr>
<td>CS30306</td>
<td>Database Management Systems (Laboratory Course)</td>
<td>139</td>
</tr>
<tr>
<td>CS31319</td>
<td>Object Oriented Modeling and Design (Laboratory Course)</td>
<td>141</td>
</tr>
<tr>
<td>CS37402</td>
<td>Mini Project</td>
<td>143</td>
</tr>
<tr>
<td>CS30401</td>
<td>Comprehensive Viva Voce</td>
<td>-</td>
</tr>
<tr>
<td>CS37301</td>
<td>Seminar</td>
<td>144</td>
</tr>
</tbody>
</table>

## Course Structure - Module VII

### Course Syllabi for Courses - Module VII

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40115</td>
<td>Data Acquisition Systems (Theory Course)</td>
<td>148</td>
</tr>
<tr>
<td>CS40114</td>
<td>Business Intelligence and Analytics (Theory Course)</td>
<td>150</td>
</tr>
<tr>
<td>*Elective Group 1 (Theory Course)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>CS42105</td>
<td>Mobile Computing</td>
<td>155</td>
</tr>
<tr>
<td>CS42127</td>
<td>Cloud Computing</td>
<td>157</td>
</tr>
<tr>
<td>CS42103</td>
<td>Parallel Computing on GPU</td>
<td>160</td>
</tr>
<tr>
<td>CS42101</td>
<td>Advanced Computer Graphics</td>
<td>163</td>
</tr>
<tr>
<td>CS42131</td>
<td>Enterprise Systems</td>
<td>165</td>
</tr>
<tr>
<td>CS42135</td>
<td>Digital Image Processing</td>
<td>168</td>
</tr>
</tbody>
</table>

12.4 *Elective Group II (Theory Course)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td>170</td>
</tr>
<tr>
<td>CS42136</td>
<td>Distributed Computing</td>
<td>173</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td>176</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td>178</td>
</tr>
</tbody>
</table>

12.5 CS40315 Data Acquisition Systems (Laboratory Course) 181

12.6 *Elective Group II (Tutorial Course)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td>182</td>
</tr>
<tr>
<td>CS42136</td>
<td>Distributed Computing</td>
<td>184</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td>186</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td>187</td>
</tr>
</tbody>
</table>

12.7 CS40314 Business Intelligence and Analytics (Tutorial Course) 189

12.8 *Elective Group I (Laboratory Course)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42305</td>
<td>Mobile Computing</td>
<td>191</td>
</tr>
<tr>
<td>CS42327</td>
<td>Cloud Computing</td>
<td>193</td>
</tr>
<tr>
<td>CS42303</td>
<td>Parallel Computing on GPU</td>
<td>194</td>
</tr>
<tr>
<td>CS42301</td>
<td>Advanced Computer Graphics</td>
<td>195</td>
</tr>
<tr>
<td>CS42331</td>
<td>Enterprise Systems</td>
<td>197</td>
</tr>
<tr>
<td>CS42335</td>
<td>Digital Image Processing</td>
<td>199</td>
</tr>
</tbody>
</table>

12.9 CS47303 Project Stage - II 200

13 Course Structure - Module VIII 203

14 Course Syllabi for Courses - Module VIII 204

14.1 CS40112 Software Testing and Quality Assurance (Theory Course) 205

14.2 CS40114 Information Systems Security (Theory Course) 208

14.3 *Elective Group III (Theory Course)

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42120</td>
<td>Data Mining</td>
<td>211</td>
</tr>
<tr>
<td>CS42128</td>
<td>Machine Learning</td>
<td>214</td>
</tr>
<tr>
<td>CS42104</td>
<td>Neural Networks</td>
<td>217</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Credits</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>CS42124</td>
<td>Algorithmic Number Theory and Algebra</td>
<td>220</td>
</tr>
<tr>
<td>CS42118</td>
<td>Geographical Information Systems</td>
<td>224</td>
</tr>
<tr>
<td>CS42129</td>
<td>Management Information Systems</td>
<td>227</td>
</tr>
<tr>
<td>CS42121</td>
<td>Advanced Computer Architecture</td>
<td>230</td>
</tr>
<tr>
<td>CS42116</td>
<td>Convergence Technologies</td>
<td>233</td>
</tr>
<tr>
<td>CS42130</td>
<td>Embedded Systems</td>
<td>235</td>
</tr>
</tbody>
</table>

Elective Group IV (Theory Course)

14.5

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42120</td>
<td>Data Mining</td>
<td>237</td>
</tr>
<tr>
<td>CS42128</td>
<td>Machine Learning</td>
<td>238</td>
</tr>
<tr>
<td>CS42104</td>
<td>Neural Networks</td>
<td>240</td>
</tr>
<tr>
<td>CS42124</td>
<td>Algorithmic Number Theory and Algebra</td>
<td>242</td>
</tr>
</tbody>
</table>

Elective Group III (Tutorial)

14.6

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42118</td>
<td>Geographical Information Systems</td>
<td>244</td>
</tr>
<tr>
<td>CS42129</td>
<td>Management Information Systems</td>
<td>245</td>
</tr>
<tr>
<td>CS42121</td>
<td>Advanced Computer Architecture</td>
<td>247</td>
</tr>
<tr>
<td>CS42116</td>
<td>Convergence Technologies</td>
<td>248</td>
</tr>
<tr>
<td>CS42130</td>
<td>Embedded Systems</td>
<td>249</td>
</tr>
</tbody>
</table>

Software Testing and Quality Assurance (Laboratory Course)

14.7

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40312</td>
<td>Software Testing and Quality Assurance</td>
<td>251</td>
</tr>
</tbody>
</table>

Information Systems Security (Laboratory Course)

14.8

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40314</td>
<td>Information Systems Security</td>
<td>253</td>
</tr>
</tbody>
</table>

Project Stage - III

14.9

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS47308</td>
<td>Project Stage - III</td>
<td>255</td>
</tr>
</tbody>
</table>

Course Syllabi for PD Courses in TY B.Tech (Information Technology)

15

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS33303</td>
<td>Advanced Java</td>
<td>259</td>
</tr>
<tr>
<td>CS33312</td>
<td>PIC Microcontroller</td>
<td>261</td>
</tr>
<tr>
<td>CS33313</td>
<td>Mobile Application Development</td>
<td>263</td>
</tr>
<tr>
<td>CS33306</td>
<td>Ethical Hacking and Network Defense</td>
<td>265</td>
</tr>
<tr>
<td>CS33310</td>
<td>Spring Framework</td>
<td>267</td>
</tr>
<tr>
<td>CS33314</td>
<td>Struts Framework</td>
<td>269</td>
</tr>
<tr>
<td>CS33311</td>
<td>Problem Solving and Programming</td>
<td>271</td>
</tr>
<tr>
<td>CS33315</td>
<td>Big Data Technologies</td>
<td>273</td>
</tr>
<tr>
<td>CS33307</td>
<td>Matlab</td>
<td>275</td>
</tr>
</tbody>
</table>
Program Educational Objectives (PEO)
B.Tech (Information Technology)

List of Programme Education Outcomes [PEO]

<table>
<thead>
<tr>
<th>PEO</th>
<th>PEO Focus</th>
<th>PEO Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEO1</td>
<td>Preparation</td>
<td>To prepare the students with a commitment towards meeting the needs of users within an organizational and societal context through the selection, creation, application, integration and administration of Information Technology projects.</td>
</tr>
<tr>
<td>PEO2</td>
<td>Core competence</td>
<td>To facilitate students with foundation of mathematical &amp; engineering fundamentals along with knowledge of Information Technology principles and applications and be able to integrate this knowledge in a variety of business and inter-disciplinary setting.</td>
</tr>
<tr>
<td>PEO3</td>
<td>Breadth</td>
<td>To enable student to exercise problem solving capacity with effective use of analysis, design, development that address idea realization.</td>
</tr>
<tr>
<td>PEO4</td>
<td>Professionalism</td>
<td>To inculcate students with professional and ethical values communication and collaboration skill and involvement in team work as a member having multidisciplinary knowledge useful to the society.</td>
</tr>
<tr>
<td>PEO5</td>
<td>Learning Environment</td>
<td>To provide students an academic environment that developed leadership qualities, excellent in subject area of computer engineering and lifelong learning in every sphere of their life.</td>
</tr>
</tbody>
</table>
List of Programme Outcomes [PO]

Graduates will be able

<table>
<thead>
<tr>
<th>PO</th>
<th>Graduate Attributes</th>
<th>PO Statement</th>
</tr>
</thead>
</table>
| P01 | **GA: 1** Engineering Knowledge | 1. To apply scientific and mathematical principles in order to determine conceptual aspects of real world problems in information engineering.  
2. To apply algorithmic principles and information science theory for comprehending technological trade-off.  
3. To explore conceptual paradigms with incorporation of programming practices. |
| P02 | **GA: 2** Problem Analysis | 4. To recognize and synthesize the context of the problem leading to correct and consistent requirements.  
5. To analyze and formulate problem frames in order to receive decomposition structure of information engineering/technology problem.  
6. To identify resources, infrastructure and technology required to realize solution of real world problem. |
| P03 | **GA: 3** Design/ Development of solution | 7. To plan and devise design alternatives which leads to conceive optimal solution.  
8. To compose technical design specifications for formally expressing the solution implementation.  
9. To practice template based approaches for formulating engineering artifacts addressing information engineering/technology problem. |
| P04 | **GA: 4** Conduct Investigation of Complex Problem | 10. To apply research knowledge in order to recognize information engineering/technology problem issues.  
11. To investigate real world information engineering/technology problem with cause effect analysis and inference.  
12. To apply research methods to determine impact and severity of |
<table>
<thead>
<tr>
<th>PO5</th>
<th>GA: 5</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modern Tool Usage</td>
<td>13. To select appropriate tools for solution development.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14. To demonstrate ability to formulate and answer empirical questions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15. To apply techniques and methods to create, enhance, and deliver IT tools.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO6</th>
<th>GA: 6</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Engineer and Society</td>
<td>16. To devise engineering solutions in a meaningful and useful way to address societal needs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17. To impart technological solutions with legal commitments and cultural diversity.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO7</th>
<th>GA: 7</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment and sustainability</td>
<td>18. To recognize impact of engineering solutions on society.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19. To adapt to changing technological scenarios in order to realize socio-technical solutions.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO8</th>
<th>GA: 8</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethics</td>
<td>20. To adhere to ethical responsibility.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21. To follow norms of engineering practice.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO9</th>
<th>GA: 9</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual and Team Work</td>
<td>22. To interact professionally at work places with effective teamwork.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>23. To demonstrate synergistic leadership skills while addressing multi-disciplinary complex problems.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO10</th>
<th>GA: 10</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>24. To demonstrate proficiency in technical and social communications.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25. To interpret and represent engineering artifacts considering IT environment.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO11</th>
<th>GA: 11</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifelong Learning</td>
<td>26. To undertake refresher courses and consultancy projects with participation in continuous development of organization.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>27. To improve the skills in refining and updating information engineering knowledge base.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28. To strive for continuous career building in information technology by higher education.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO12</th>
<th>GA: 12</th>
<th>Problem statement:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Management</td>
<td>29. To determine project specific constraints, forces, resources and schedule.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30. To acquire projects from competitive global world by</td>
<td></td>
</tr>
<tr>
<td>and Finance</td>
<td>conceptualizing and framing unique ideas.</td>
<td></td>
</tr>
</tbody>
</table>
### Module 1

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L   P  Tut.</td>
<td>ISA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test 1  Test 2  HA  Tut.  CA</td>
<td>ESE</td>
<td></td>
</tr>
<tr>
<td>HS10109</td>
<td>Linear Algebra and Random Variables</td>
<td>Theory - Core</td>
<td>3 - 1</td>
<td>10  20  5  5 -</td>
<td>60  4</td>
</tr>
<tr>
<td>HS10103</td>
<td>Modern Physics</td>
<td>Theory – Core</td>
<td>3 - 1</td>
<td>10  20  5  5 -</td>
<td>60  4</td>
</tr>
<tr>
<td>CH10101</td>
<td>Chemistry</td>
<td>Theory – Core</td>
<td>3 - -</td>
<td>15  20  5 - -</td>
<td>60  3</td>
</tr>
<tr>
<td>ME10101</td>
<td>Engineering Graphics</td>
<td>Theory – Core</td>
<td>3 - -</td>
<td>15  20  5 - -</td>
<td>60  3</td>
</tr>
<tr>
<td>HS16101</td>
<td>Sociology</td>
<td>Theory – OE</td>
<td>2 - -</td>
<td>15  20  5 - -</td>
<td>60  2</td>
</tr>
<tr>
<td>HS16103</td>
<td>Psychology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS16105</td>
<td>Philosophy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS16107</td>
<td>EVS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS10301</td>
<td>Engineering Graphics Lab</td>
<td>Lab – Core</td>
<td>- 2 -</td>
<td>- - - -</td>
<td>70  30 1</td>
</tr>
<tr>
<td>CH10301</td>
<td>Science Lab</td>
<td>Lab – Core</td>
<td>- 2 -</td>
<td>- - - -</td>
<td>70  30 1</td>
</tr>
<tr>
<td>HS17401</td>
<td>Mini Project</td>
<td>Project</td>
<td>- 4 -</td>
<td>- - - -</td>
<td>70  30 2</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>14  08  2</td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>
## Module 2

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L</td>
<td>P</td>
<td>Tut.</td>
<td>ISA</td>
</tr>
<tr>
<td>HS10102</td>
<td>Differential Integral and Calculus</td>
<td></td>
<td></td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>HS10104</td>
<td>Engineering Mechanics</td>
<td></td>
<td></td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>HS10108</td>
<td>Electrical Engineering Fundamentals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS10102</td>
<td>Computer Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS16102</td>
<td>Economics Management Technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS16104</td>
<td>Cost &amp; Acc. Business Law</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS10302</td>
<td>Computer Programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS10306</td>
<td>Engineering Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS17402</td>
<td>Mini Project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL** | 14 | 08 | 2 | 20 |
### F.Y. B. Tech. Structure with effect from Academic Year 2015-16

#### Semester I – Irrespective of Module

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>P</td>
<td>Tut.</td>
</tr>
<tr>
<td>HS10107</td>
<td>Communication Skill</td>
<td>Comm. Skill</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>HS153xx</td>
<td>General Proficiency</td>
<td>GP</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>HS15301</td>
<td>English I</td>
<td>Language</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>HS15302</td>
<td>French I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS15303</td>
<td>German I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS15304</td>
<td>Spanish I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS15305</td>
<td>Japanese I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS14301</td>
<td>Engineering Workshop</td>
<td>Workshop</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
</tbody>
</table>
### F.Y. B. Tech. Structure with effect from Academic Year 2015-16

#### Semester II – Irrespective of Module

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject</th>
<th>Type</th>
<th>Teaching Scheme</th>
<th>Assessment Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>P</td>
<td>Tut.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test 1</td>
<td>Test 2</td>
<td>HA</td>
</tr>
<tr>
<td>HS17301</td>
<td>General seminar I</td>
<td>Comm.</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>HS153xx</td>
<td>General Proficiency</td>
<td>GP</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>HS15306</td>
<td>French II</td>
<td>Language</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>German II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spanish II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japanese II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>English II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS14302</td>
<td>Trade Workshop</td>
<td>Workshop</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
</tbody>
</table>

**HS153xx : General Proficiency Courses as per following list**

### List of General Proficiency Courses

**FY B Tech**

**AY 2015-16**

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Course Code</th>
<th>Name of Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HS15311</td>
<td>Flute</td>
</tr>
<tr>
<td>2</td>
<td>HS15312</td>
<td>Guitar</td>
</tr>
<tr>
<td>3</td>
<td>HS15313</td>
<td>Tabla</td>
</tr>
<tr>
<td>4</td>
<td>HS15314</td>
<td>Bharat Natyam</td>
</tr>
<tr>
<td>5</td>
<td>HS15317</td>
<td>Yoga</td>
</tr>
<tr>
<td>6</td>
<td>HS15318</td>
<td>Pranayam</td>
</tr>
<tr>
<td>7</td>
<td>HS15319</td>
<td>Aerobics</td>
</tr>
<tr>
<td>8</td>
<td>HS15321</td>
<td>Photography</td>
</tr>
<tr>
<td>9</td>
<td>HS15322</td>
<td>Digital Photography</td>
</tr>
<tr>
<td>10</td>
<td>HS15323</td>
<td>Volleyball</td>
</tr>
<tr>
<td>11</td>
<td>HS15324</td>
<td>Chess</td>
</tr>
<tr>
<td>12</td>
<td>HS15325</td>
<td>Taekwondo</td>
</tr>
<tr>
<td>13</td>
<td>HS15326</td>
<td>Film Appreciation</td>
</tr>
<tr>
<td>14</td>
<td>HS15327</td>
<td>Shares and Stocks</td>
</tr>
<tr>
<td>15</td>
<td>HS15328</td>
<td>Fundamentals of Banking</td>
</tr>
<tr>
<td>16</td>
<td>HS15329</td>
<td>Nutrition and Fitness</td>
</tr>
<tr>
<td>17</td>
<td>HS15330</td>
<td>Spirit of Entrepreneurship</td>
</tr>
<tr>
<td>18</td>
<td>HS15331</td>
<td>Memory Techniques</td>
</tr>
<tr>
<td>19</td>
<td>HS15332</td>
<td>Sanskrit Sambhashanam</td>
</tr>
<tr>
<td>20</td>
<td>HS15333</td>
<td>Numerology</td>
</tr>
</tbody>
</table>
CS10102:: COMPUTER PROGRAMMING

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1 : Introduction to Programming (8+2 Hrs)
PART A: Problem solving using computers and logic design. Algorithms and their representations: flowcharts, pseudo code. Designing algorithms for problems like finding min-max, mean, median, mode, mensuration and roots of a quadratic equation. Concept of programming languages for implementing algorithms – levels of languages. Role of assemblers, compilers, linker, loader, interpreter in program execution.


Unit 2: Flow of Control (8+2 Hrs)

Arrays: Concept, declaration and initialization of arrays, accessing individual elements of array. Use of arrays in sorting, searching. Concept of 2-D array (Matrix), row major and column major representation of array, address calculation for accessing the individual element.

PART B. Static variables and constants in C language.

Unit 3: Functions (8+2 Hrs)
PART A: Need of functions, function declaration, definition and call. Inbuilt functions and user defined functions. Passing arguments to a function, returning values from a function. Scope of variable, local and global variable. Access specifiers. Passing arrays to functions.


PART B: Preprocessor and preprocessor directives: macro substitution, difference between macro and functions.
Unit 4: Pointers and Strings (8+2 Hrs)


Strings: Strings as arrays, character array versus strings, reading strings, writing strings, user defined functions for string operations — copy, concatenate, length, reverse, converting case, appending, comparing two string, extracting a substring. Array of strings.

PART B: Const keyword in C, standard string library functions in string.h for string manipulation.

Unit 5: Structures and File Handling (8+2 Hrs)

PART A: Structures: Notion, declaration and initialization, structure variables, accessing and assigning values of the fields, "size of" operator, functions and structures, arrays of structures, nested structures, pointers and structures, passing structure to a function and returning structure from function. Dynamic memory allocation, type casting, Introduction to self referential structures, linked list as a dynamic alternative to arrays.

File Handling in C: file types, file opening modes, file handling I/O – fprintf, fscanf, fwrite, fread, fseek. File pointers. Implementing basic file operations in C.


Text Books

Reference Books

Additional Reading

Course Outcomes:

Upon completion of the course, graduates will be able to -
1. List procedural programming benefits to construct concise solutions
2. Interpret and develop naturo-visual representation of problem in hand.
3. Apply available algorithmic principles to general efficient solutions
4. Justify modular programming approach by making use of elementary as well as superior data structures.
5. Apply programming fundamentals with generic prototype.
6. Evaluate and manipulate given solutions in reengineered view
CS10302: COMPUTER PROGRAMMING LAB

Credits: 01

Teaching Scheme: Laboratory 2 Hrs/Week

Prerequisites:

List of Practicals

1. Study of most important DOS/UNIX commands.

2. Write a program in C to find largest element / average of given N elements / sum / reverse of a given integer.

3. Write a program in C to implement a simple mathematical calculator

4. Write a program in C to read an integer and display each of the digits of an integer in English.

5. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

6. Write a program in C to perform Addition / Subtraction / Multiplication of two Matrices. Also determine whether the matrix is symmetric / skewed.

7. Write a program in C to carry out following operations on strings using string library Functions:
   a. Length of a string.
   b. Copy a string.
   c. Concatenation of strings.

8. Write a program in C to carry out following operations on strings without using string library functions
   a. Compare two strings.
   b. Reverse given string.
   c. To check if the given string is a palindrome or not.

9. Write a program in C to carry out following operations on strings using pointers.
   a. Length of a string.
   b. Concatenation of strings.
   c. Copy of string
   d. Compare two strings.

10. Write a C program that works with complex numbers using a structure. Perform the
following operations:
   a. Reading a complex number.
   b. Addition of two complex numbers.
   c. Writing a complex number.
   d. Multiplication of two complex numbers.

11. Write a C program to create a database of students by using array of structure and perform following operations on it.
   a. Accept/modify record of student
   b. Search a particular record
   c. Display all records

12. Write a program in C that use both recursive and non-recursive functions to find the Factorial / GCD (greatest common divisor) of two given integers / Fibonacci series.

13. Write a program in C to sort n integers using bubble / merge sort.

14. Write a program in C to search a number in a given list using linear / binary search.

Text Books

Reference Books

Additional Reading
MODULE III
## Course Structure

**Branch** - Computer Engineering  
**Year** - Third Year of Engineering  
**Semester** - 1  
**Year of Engineering** - 2015-2016  
**Pattern** - f11

### Theory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs./week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS20117</td>
<td>Discrete Structures and Graph Theory</td>
<td>3 Lectures, 0 Tutorial</td>
<td>HA: 5 - Test 1: 10, Test 2: 20, ESE: 60, Total: 100</td>
<td>3</td>
</tr>
<tr>
<td>CS20111</td>
<td>Data Structures</td>
<td>3 Lectures, 0 Tutorial</td>
<td>HA: 5 - Test 1: 15, Test 2: 20, ESE: 60, Total: 100</td>
<td>3</td>
</tr>
<tr>
<td>CS20108</td>
<td>Computer Organization</td>
<td>3 Lectures, 0 Tutorial</td>
<td>HA: 5 - Test 1: 15, Test 2: 20, ESE: 60, Total: 100</td>
<td>3</td>
</tr>
<tr>
<td>CS20116</td>
<td>Problem Solving and Programming</td>
<td>2 Lectures, 1 Tutorial</td>
<td>HA: 5 - Test 1: 10, Test 2: 20, ESE: 60, Total: 100</td>
<td>3</td>
</tr>
<tr>
<td>*CS20113</td>
<td>Digital Electronics and Logic Design</td>
<td>3 Lectures, 1 Tutorial</td>
<td>HA: 5 - Test 1: 10, Test 2: 20, ESE: 60, Total: 100</td>
<td>4</td>
</tr>
<tr>
<td>#CS20114</td>
<td>Microprocessor and Interfacing</td>
<td>2 Lectures, 1 Tutorial</td>
<td>HA: 5 - Test 1: 10, Test 2: 20, ESE: 60, Total: 100</td>
<td>4</td>
</tr>
</tbody>
</table>

**TOTAL**  
14 Lectures, 2 Tutorials  
16 Credits

### Laboratory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs./week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS20311</td>
<td>Data Structures</td>
<td>4 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>2</td>
</tr>
<tr>
<td>#CS20314</td>
<td>Microprocessor and Interfacing</td>
<td>2 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>1</td>
</tr>
<tr>
<td>*CS20313</td>
<td>Digital Electronics</td>
<td>2 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>1</td>
</tr>
<tr>
<td>CS27401</td>
<td>Mini Project</td>
<td>4 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>2</td>
</tr>
<tr>
<td>CS24306</td>
<td>PHP MYSQL OR</td>
<td>2 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>1</td>
</tr>
<tr>
<td>CS24303</td>
<td>C#.NET</td>
<td>2 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>1</td>
</tr>
<tr>
<td>*LAB3</td>
<td>Technical Writing</td>
<td>2 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>1</td>
</tr>
<tr>
<td>#LAB3</td>
<td>General Seminar-2</td>
<td>2 Practical</td>
<td>ISE: 30, ESE: 70, Total: 100</td>
<td>1</td>
</tr>
</tbody>
</table>
### Abbreviations Used

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>Home Assignment</td>
<td>2</td>
</tr>
<tr>
<td>TA</td>
<td>Teacher Assessment</td>
<td>9</td>
</tr>
<tr>
<td>ISE</td>
<td>In Sem. Examination</td>
<td></td>
</tr>
<tr>
<td>ESE</td>
<td>End Sem. Examination</td>
<td></td>
</tr>
<tr>
<td>CA</td>
<td>Continuous Assessment</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Alternate Week Laboratory</td>
<td></td>
</tr>
<tr>
<td>^</td>
<td>Detail Enclosed Separately</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Students will register only in Semester III</td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>Students will register only in Semester IV</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL CREDITS**: 25

---

CS20117:: DISCRETE STRUCTURES AND GRAPH THEORY

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Elementary knowledge of 1) Numbers 2) Probability

Unit 1 : Logic and Proofs
(8 hrs)
Part A: Propositional logic, propositional equivalences, predicates and quantifiers, rules of inference, introduction to proofs- direct, trivial, contraposition, contradiction, counterexamples; Normal forms – DNF and CNF.
Part B: Program Correctness

Unit 2 Elementary Discrete Structures & Basic Counting
(8 hrs)
Part A: Elementary set theory, relations, functions, principle of mathematical induction, basic counting principles, permutations, combinations, generalized permutations and combinations (with/without repetitions), Permutations with indistinguishable objects, Binomial coefficients and identities.
Part B: Generalized permutations and combinations (distinguishable/indistinguishable objects)

Unit 3 : Advanced Counting Techniques
(8 hrs)
Part A: Pigeon-Hole Principle : Some elegant applications, Inclusion Exclusion Principle : Counting with Venn Diagrams, (some examples from counting Derangements, number of primes upto n, number of onto functions, Euler’s phi function), Recurrence relations, modeling using recurrence relations (some examples from: Fibonacci numbers, Catlan numbers, Derangements, Tower of Hanoi), solution of linear recurrence relations with constant coefficients (homogenous and non-homogenous), generating functions and coefficients.
Part B: Turan’s generalization of Mantel’s theorem, Mobius inversion formula and some applications.

Unit 4 : Modular Arithmetic
(8 hrs)
Part A : Number theory – Division Algorithm, Euclid’s Algorithm, extended Euclid’s algorithm, modular inversion, Fundamental Theorem of Arithmetic, Congruence’s, Fermat’s little theorem, Euler’s phi function, Chinese remainder theorem, Diffie-Hellman and RSA algorithms.
Part B : Fast exponentiation

Unit 5 : Graph Theory
(8 hrs)
Part A : Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, connected components, degree of a vertex, paths,
cycles in graph, Euler and Hamiltonian tours/graphs, Trees, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Theorem on bound on number of edges, Graph colorings

**Part B:** Hall's marriage theorem, perfect matching’s in graph, Tutte’s theorem, Konig’s theorem,

**Text Books:**

1. *Discrete Mathematics and its applications* by Kenneth Rosen (William C Brown Publisher)
2. *Applied Combinatorics* by Alan Tucker (Wiley Publishing company)
3. *Combinatorics: Topics, techniques, algorithms* by Peter J. Cameron (Cambridge University Press)
4. *Graph Theory* by Reinhard Diestel (Springer Verlag Publishing Company)
5. *Introduction to Graph Theory* by Douglas B. West (Prentice-Hall publishers)
7. *Elementary Number Theory*, David Burton

**Reference Books:**

2. *Algebra* by Michael Artin (Pearson Prentice Hall)

**Course Outcomes:**

Upon completion of the course, graduates will be able to –

1. Reason mathematically on elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems
2. Describe the elementary properties of modular arithmetic and their applications in Computer Science like cryptography.
3. Summarize graph theory fundamentals and their applications
4. Develop recurrence relations for a wide variety of interesting problems
5. Express mathematical properties formally via the formal language of propositional logic and predicate logic
6. Demonstrate use of pigeon-hole and inclusion-exclusion principle in solving elegant or important problems
CS2011: Data Structures

Credits: 03
Teaching Scheme: - Theory 3 Hrs/Week
Prerequisites:

Unit 1: Fundamentals of Data Structures, Sorting & Searching (8 Hrs)


Part B: External Sorting, Sparse Matrix: Addition and Fast transpose

Unit 2: Elementary Data Structures (8 Hrs)


Part B: Priority Queue

Unit 3: Linked Lists (8 Hrs)

Part A: Singly Linked Lists, Doubly linked Lists, Circular linked lists, Generalized linked lists, Applications: Stack & Queue using linked list, Polynomial Manipulation using linked list & Generalized linked list.

Part B: Dynamic memory allocation for matrices and operations on matrices, Sparse matrix representation

Unit 4: Trees (8 Hrs)

Part A: Basic terminology, representation using array and linked list, Tree Traversals: Recursive and Non recursive, Operations on binary tree: Finding Height, Leaf nodes, counting no of nodes etc., Construction of binary tree from traversals, Binary Search trees (BST): Insertion, deletion of a node from BST, Optimal Binary Search tree (OBST), Threaded Binary tree (TBT): Creation and traversals on TBT, Height Balanced Tree (AVL): Rotations on AVL tree, M-way search trees: Btrees, B⁺ tree

Part B: Red-Black Trees, Game trees, Expression tree

Unit 5: Graph (8 Hrs)

Part A: Terminology and representation, Traversals, Connected components and Spanning trees: Prim’s and Kruskal’s Algorithm, Shortest Paths and Transitive Closures:
Single Source all destinations (Dijkstra’s Algorithm), All pair Shortest Path Algorithm, Activity Network, Topological Sort and Critical Path

**Part B :** Multistage Graphs

**Text Books:**

**Reference Books :**

**Course Outcomes:**
Upon completion of the course, graduates will be able to -

1. To interpret and diagnose the properties of data structures with their memory representations.
2. To handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. To use linear and non-linear data structures like stacks, queues, linked list etc.
4. To demonstrate the use of binary tree traversals.
5. To analyze the given problem in terms of complexity.
6. To use an appropriate data structure and algorithm to solve a problem.
CS20108:: COMPUTER ORGANIZATION

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit I : Structure of a Computer System  
( 8 Hrs )


Part B: PCI Bus. IEEE standards for Floating point representations (Double Precision Format).

Unit 2: Processor Organization (CPU)  
( 8 Hrs )


Part B: Introduction to i7 processor.

Unit 3: Control Unit  
( 8 Hrs)


Part B: Multiple-bus organization. Applications of micro programming.

Unit 4: Memory Organization  
( 8 Hrs)


Part B: Secondary Storage: Magnetic Disk, Optical memory, CDROM, RAID

Unit 5: Advanced Computer Organizations  
( 8 Hrs)

Part A: The Intel IA-64 Architecture: General Organization, Prediction, Speculation Software pipelining, Instruction set architecture. Introduction to parallel processing: Trends towards parallel processing, architectural classification schemes (Flynn’s classification) Introduction to multicore processor AMD/ NVIDIA GPU architecture.

Part B: Itanium Organization

Text Books

Reference Books

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Recognize historical scenario of computing units development with regards to technological evolution.
2. Analyze design tradeoffs pertaining to performance and memory issues.
3. Access impact of circuit-driven and program driven analogies to assemble realizable solutions.
4. Operationalize and judge arithmetic and control units based on architectures proportionate to indicate I/O processing.
5. Suggest performance-bound solutions in order to demonstrate variety of technologies.
6. Represent arithmetic-control aspects adhering to standard I/O devices and interfaces.
CS20116:: PROBLEM SOLVING AND PROGRAMMING

Credits: 02

Teaching Scheme: - Theory 2 Hrs/Week

Prerequisites: Computer Programming

Unit 1: Introduction (6+1 Hrs)
PART A: General Problem solving techniques: Examples of problems that are solved using different approaches: ask questions, look for things that are familiar, solve by analogy, means-ends analysis, divide and conquer (top-down approach or stepwise refinement), building block approach, merging solutions, working backwards from a solution, binary doubling strategy, iterative vs recursive solutions, parallel techniques
Program Verification: computer model for program execution, correctness of programs, input and output assertions, implication and symbolic execution, verification of different types of program segments – straight line program segment, having branch, having loops, loop invariants, using arrays. Proof of termination. Debugging programs and program testing. The challenge of Binary Search.
Efficiency of programs: considerations during design, considerations during implementation, reducing time and space requirements of programs, choosing right data structures and data types – arrays, lists, tables, bitmaps. When to use what?
PART B: Input validation – GiGO, input validation loop, defensive programming

Unit 2: Basic Algorithms for Integers (6+1 Hrs)
PART A: Fundamental Algorithms – exchanging values of two variables, generating Fibonacci sequence, reversing digits of an integer, base conversion, character to number conversion
Factoring Methods: finding square root of a number, finding GCD of two numbers, generation of pseudo random numbers, generating prime numbers, finding prime factors of an integer, raising a number to a large power.
PART B: Simple modular arithmetic - last k digits of number a^b, finding recurring decimal expansion for rational number a/b

Unit 3: Numerical Methods (6+1 Hrs)
PART B: Simpson’s rule, Trapezoidal rule

Unit 4: Recursion and Arrays (6+1 Hrs)
Array Techniques: array order reversals, removal of duplicates from an ordered array, partitioning an array, finding the kth smallest element, largest monotone subsequence. Using parallel arrays and higher dimensional arrays. Algebraic equations - Gaussian elimination

PART B: 8 queens problem, Gauss-Siedel method

Unit 5: Text Processing (6+1 Hrs)

PART A: String processing and pattern searching: text line length adjustment, left and right justification of text, keyword searching in text, text line editing, linear pattern search, sublinear pattern search, character-by-character text processing. Some standard text processing problems.

PART B: Big number arithmetic using strings for representing numbers – for example multiplying 100 digit numbers or finding factorial of 100

Text Books
2. “How to Solve It by Computer”, R. G. Dromey, PHI
3. “Starting out with programming logic and design”, 3rd edition, Tony Gaddis, Pearson publications
4. “Programming Logic and Design Introductory”, sixth edition, Joyce, Farrell, Course Technology, CENGAGE Learning
5. “An Introduction to programming through C++”, Abhiram Ranade, McGraw Hill Education

Reference Books

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Apply and practice logical ability to solve the problems.
2. Modularize the problems into small modules and then convert them into algorithms
3. Analyze algorithms and determine their time complexity.
4. Trace and code recursive programs.
5. Choose appropriate problem solving technique
6. Verify and validate the correctness of the algorithm.
CS20113: DIGITAL ELECTRONICS AND LOGIC DESIGN

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1 : Number system and Codes (8+1 Hrs)
PART B: Minimization of POS forms using K- Map, Quine-McCluskey Method, Half adder and subtractor, full adder and subtractor.

Unit 2: Combinational & Sequential logic circuits (9+1 Hrs)
PART B: ALU 74181, BCD-to-7-segment Decoder, adder with look ahead carry generator.

Unit 3 : Design of sequential circuits (8+1 Hrs)
PART A: Moore/Mealy M/c's: representation techniques, state diagrams, state tables, state reduction, state assignment, implementation using flip-flops. Applications like sequence generator and detector. Shift register (modes of operation), 4 bit bi-directional universal shift register, application of shift registers (Ring counter, Sequence generator, Johnson's counter.)ASM charts, notations, design of simple controller, multiplexer controller method, RTL notations and implementation.
PART B: Examples on ASM, RTL.

Unit 4 : Logic Families (8+1 Hrs)
PART A: Characteristics of Digital ICs: Speed, Power dissipation, fan-out, current and voltage parameters, noise margin, operating temperature etc., TTL: Operation of TTL NAND gate, Standard TTL, TTL Characteristics, Active pull-up, Wired-AND, totem pole, open collector, Unconnected Inputs. CMOS Logic: CMOS Inverter, CMOS NAND and NOR, CMOS characteristics. Wired-logic, Unconnected Inputs, Open-Drain Outputs, Comparison of TTL and CMOS, interfacing TTL to CMOS

PART B: Interfacing CMOS to TTL, Tri-state logic: tri-state buffers, inverters, Study of Data sheets of 7400 Series ICs: (Basic and Universal logic gates)

Unit 5: Programmable Logic Devices (8+1 Hrs)

PART A: Programmable Logic array: Input, Output Buffers, AND, OR, Invert/Non-Invert Matrix, Programming the PLA, Applications of PLAs to implement combinational and sequential logic circuits Introduction to :FPGA, CPLD.

Introduction to VHDL: Modeling Digital systems, modeling languages, modeling concepts.

PART B: Comparison of FPGA and CPLD, VHDL Programs.

Text Books

Reference Books

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Minimize logical equations
2. Design code convertors
3. Construct Combinational and Sequential circuits
4. Validate design outputs using standards test equipments
5. Develop applications of sequential circuits
6. Describe Programmable Logic Devices
CS20216:: PROBLEM SOLVING AND PROGRAMMING

Credits: 01  Teaching Scheme: -Tutorial 1Hr/Week

Prerequisites: Computer Programming

List of Contents

A TERM-WORK containing the record of the following:

1. Implementing logic to draw basic shapes on computer using some library to take care of drawing details (basic shapes like square, circle, polygons etc., basic animation, projectile motion, bouncing ball etc.)

2. Visualizing simulations (e.g. cosmological simulation) using a library to take care of drawing details

3. Modular arithmetic (finding last k digits of number a^b, finding recurring decimal expansion for rational number a/b, etc.)

4. Factoring methods (generation of pseudo random numbers, smallest divisor of an integer, computing n^{th} Fibonacci number, etc.)

5. Taylor Series expansion (with an emphasis on program specification, testing, proof of correctness, invariants, debugging, good coding practices) taking sine, cosine, e etc. as examples

6. Applications of recursion (drawing recursive pictures, layout of mathematical formulae, finding min-max, etc.)

7. Array based techniques (Gaussian elimination, Gauss-Siedel method, etc.)

8. Text processing and pattern searching (finding LCS in strings, finding duplicate words, spell checker, etc.)

Text Books
2. “How to Solve It by Computer”, R. G. Dromey, PHI
3. “Starting out with programming logic and design”, 3rd edition, Tony Gaddis, Pearson publications
4. “Programming Logic and Design Introductory”, sixth edition, Joyce, Farrell, Course Technology, CENGAGE Learning
5. “An Introduction to programming through C++”, Abhiram Ranade, McGraw Hill Education

Reference Books
CS20113::DIGITAL ELECTRONICS AND LOGIC DESIGN

Credits: 01
Teaching Scheme: - Tutorial 1 Hr/Week
Prerequisites:

1. Introduction to Electronics Workbench/Multisim, IC 74XX and IC 54XX.
2. Design BCD to Seven Segment Display using common cathode and common
   anode display.
4. Encoder, Decoder and Priority Encoder Applications (Barrel Shifter, Realization
   of ROM).
5. Design MOD N counter using IC-74190, 74193.
6. Design Sequence detector using Moore and Mealy Model to detect given
   sequence.
7. Numerical based on RTL.
8. Numerical based on ASM.
9. Study various Characteristics of Digital ICs.
10. Design combinational circuit using ROM, PLA and PAL.
11. Write VHDL Programs for combinational and Sequential circuits.

Text Books

   Edition.

Reference Books


Additional Reading

CS20311:: DATA STRUCTURES FF No. : 654 B

Credits: 02
Teaching Scheme: - Laboratory 4 Hrs/Week
Prerequisites: Computer Programming

List of Lab Experiments

1. Implementation Quick and Merge Sort.
2. Implementation of Heap Sort
4. Simulation of Recursion using STACK.
5. Implementation of Expression conversion and Evaluation.
6. Simulation of Job Scheduling Using Queue
7. Polynomial Manipulation using Queue
8. Implementation of various operations on singly linked list.
9. Implement Generalized linked list and various operations on it.
10. Implement various operations on Doubly Linked list.
11. Implement various operations on Binary Search tree.
13. Implement Btree, B*tree and operation on it.
15. Implement BFS, DFS on Graph.
16. Implement Prim’s and Kruskal’s Algorithm.
17. Implement Dijkstra’s algorithm
18. Implement all pair shortest path problem.

Text Books:


Reference Books:

CS20313:: DIGITAL ELECTRONICS

Credits: 01  
Teaching Scheme: - Laboratory  2 Hrs/Week

Prerequisites:

List of Practical

I Combinational Logic Design

1. Verification of Logical Gates and Boolean Algebra.
2. Code converters, e.g. Excess-3 to BCD and vice versa using logical gates.
3. Multiplexer - e.g. 16:1 Mux using 4:1 Mux (IC 74153).
4. Decoder – e.g. 2 bit comparator (IC 74138).
5. BCD adder – using IC 7483

II Sequential Circuit Design (Any six)

1. Conversion of flip-flops. e.g. JK to D, T.
2. Ripple (asynchronous) mod –N counter using J-K F-F.
3. Ripple (asynchronous) mod –N counter using IC 7490.
5. Sequence generator using JK flip-flop
6. Pseudo random number generator using 74194.(universal shift register)
7. Sequence detector (Moore ckt) using JK flip-flop
8. Sequence detector (Mealy ckt) using JK flip-flop

III ASM (Any one)

2. Design of simple combinational circuit: half adder and subtractor using VHDL language.
Text Books

Reference Books

Additional Reading
CS27401: MINIPROJECT

Credits: 02

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -

1. Recognize essential & dominant area of technology for achievable artifacts over rapid period of time.
2. Acquire rapid application development cycle involving prototyping to learn adequate technological environments.
3. Concisely formulate specific problem in drafted specification format.
4. Devise data dictionaries and solution design with sufficient details.
5. Demonstrate the crafted solutions to user community with a lean learning curve.
6. Validate newer dimension of extendable and scalable nature of the problem solution crafting.
CS24306 PHP MYSQL

Credits: 01

Teaching Scheme: 2 Hours / Week

List of Practical’s:

1. Download, Install and Configure Netbeans IDE 8.x/eclipse with XAMPP/WAMP/IIS/Tomcat 6.x and MySQL server for PHP
2. Design a Web page using HTML5 and CSS.
3. Design a PHP page to demonstrate the use of variables, functions, conditional and lopping constructs.
4. Design a Web form using PHP and apply validation.
5. Design a Web page to demonstrate the use of session and cookie.
6. Implement user defined exception handling for the Web page.
7. Design a database application in PHP using MySQL.
8. Design a client agent to send an email in PHP
9. Design File upload and download program in PHP
10. Design Web page filters in PHP
11. Design Web page using AJAX and PHP
12. Mini project

Text Books:

Reference Books:

Course Outcomes:
Upon completion of the course, graduates will be able to –

1. Use MySQL database, Flat files
2. Create user defined exceptions, page filters, session and cookie
3. apply the effects of HTML5, CSS, AJAX
4. Choose suitable software tools for Web Development Application
5. Design and deploy dynamic and interactive web pages.
6. Demonstrate the Model-View-Control design pattern for Web Application.
CS24303: C#.NET

Credits: 01

Teaching Scheme: - Lab 2 Hrs/Week

Prerequisites: C programming

List of Practical
2. Implementing Arrays, Strings and System collections in C#.
3. Implementing Classes, objects, constructors in C#
4. Design a simple C#.net application for calculator.
5. Design a simple C#.net application for notepad/word pad using menu editor.
6. Handling multiple forms in C#.
7. Design Database application.
9. File handling and Exception Handling in C#
10. Publishing and Deployment of windows application in C#.net/Developing DLLs
11. Mini project.

Text Books

Reference Books

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Display proficiency in C# programming by building stand-alone applications in the .NET framework.
2. Analyze Common Language Runtime (CLR), garbage collection, and assemblies, forms, collections, constructs, delegates, events and exception handling.
3. Create data-driven applications using the .NET Framework, C# and ADO.NET
4. Design application and projects using Visual Studio IDE.
5. Evaluate problems and alternative solutions using C# in a wide variety of business and organizational contexts in different socio-cultural environments.
MODULE IV
# Course Structure

**Branch** – Computer Engineering  
**Year** - THIRD Year of Engineering  
**Semester** - 1  
**Academic Year** – 2015-2016  
**Pattern** – f11

## Theory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme ( Hrs. / week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS21104</td>
<td>Mathematical Transforms and Applications</td>
<td>3 1</td>
<td>5 5 10 20 60 100</td>
<td>4</td>
</tr>
<tr>
<td>CS20110</td>
<td>Computer Graphics</td>
<td>3 0</td>
<td>5 - 15 20 60 100</td>
<td>3</td>
</tr>
<tr>
<td>CS20105</td>
<td>Principles of Programming Languages</td>
<td>3 0</td>
<td>5 - 15 20 60 100</td>
<td>3</td>
</tr>
<tr>
<td>CS21112</td>
<td>Data Communication</td>
<td>2 0</td>
<td>5 - 15 20 60 100</td>
<td>2</td>
</tr>
<tr>
<td>*CS20113</td>
<td>Digital Electronics and Logic Design</td>
<td>3 1</td>
<td>5 5 10 20 60 100</td>
<td>4</td>
</tr>
<tr>
<td>#CS20114</td>
<td>Microprocessor and Interfacing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Microprocessor and Interfacing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL**  
14 2 16

## Laboratory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme ( Hrs. / week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS20310</td>
<td>Computer Graphics</td>
<td>2</td>
<td>30 70 100</td>
<td>1</td>
</tr>
<tr>
<td>CS20305</td>
<td>Principles of Programming Languages</td>
<td>2</td>
<td>30 70 100</td>
<td>1</td>
</tr>
<tr>
<td>#CS20314</td>
<td>Microprocessor and Interfacing</td>
<td>2</td>
<td>30 70 100</td>
<td>1</td>
</tr>
<tr>
<td>*CS20313</td>
<td>Digital Electronics</td>
<td>2</td>
<td>30 70 100</td>
<td>1</td>
</tr>
<tr>
<td>CS27402</td>
<td>Mini Project</td>
<td>4</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>CS24302</td>
<td>ASP.net OR</td>
<td>2</td>
<td>30 70 100</td>
<td>1</td>
</tr>
<tr>
<td>CS24304</td>
<td>Python</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS24307</td>
<td>Ruby Programming</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Technical Writing General Seminar-2

<table>
<thead>
<tr>
<th>Code</th>
<th>Course Description</th>
<th>Credits</th>
<th>Theory Marks</th>
<th>Practical Marks</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS20401</td>
<td>Comprehensive Viva Voce Based on Courses S2, S3, S5</td>
<td>2</td>
<td>30</td>
<td>70</td>
<td>100</td>
</tr>
</tbody>
</table>

**TOTAL**: 16

**TOTAL CREDITS**: 25

#### Abbreviations Used

- **HA**: Home Assignment
- **TA**: Teacher Assessment
- **ISE**: In Sem. Examination
- **ESE**: End Sem. Examination
- **CA**: Continuous Assessment
- **^**: Detail Enclosed Separately
- **▲**: Students will register only in Semester III irrespective of Module
- *****: Students will register only in Semester IV irrespective of Module

---

CS21104: MATHEMATICAL TRANSFORMS AND APPLICATIONS

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Engineering Mathematics- I and Engineering Mathematics - II

Unit 1: Linear Differential equations of higher order (08 Hrs)  
PART B: Electrical circuits and Coupled circuits

Unit 2: Complex Analysis (08 Hrs)  
PART A: Complex differentiation, Analytical functions, Cauchy-Riemann equations, Complex Integration, Cauchy’s Integral Theorem and formula, Residue Theorem and applications to Engineering Problems, Power series, Taylor series, Laurent series, Radius of convergence.  
PART B: Bilinear Transformations and Conformal mapping.

Unit 3: Laplace Transform (08 Hrs)  
PART B: Application of Laplace transform to simultaneous differential equations.

Unit 4: Fourier Transform (08 Hrs)  
PART A: Introduction to signals and their properties, mathematical operations on signals, Complex Fourier series and frequency spectrum, Fourier integrals, Fourier cosine and sine transforms, Fourier transforms, properties of Fourier transform, Discrete Fourier transform, Properties.  
PART B: Applications of FT and DFT

Unit 5: Z Transform (08 Hrs)  
PART B: Difference equations by Z transform method.
Text Books


Reference Books

1. Michael D. Greenberg; Advanced Engineering Mathematics; Pearson Education Asia
2. Dennis G. Zill, Michael R. Cullen; Advanced Engineering Mathematics; Narosa Publishing House
4. Robert A. Gabel, Richard A. Roberts; Signals and linear systems; John Wiley & sons.
UNIT 1: BASIC CONCEPTS

PART A: Graphics Primitives: Introduction to computer graphics, Display adapters, Display modes, Pixel, Frame Buffer, Display file structure, Display file interpreter, Raster scan & random scan displays, Aspect Ratio.

Mathematical foundations: Lines and line representations, Vectors, Intersection of lines, Normalized Device Coordinates.

Scan conversions: DDA and Bresenham’s line drawing algorithms and Bresenham and Midpoint circle drawing algorithms, Arcs and Sectors, Aliasing and Antialiasing, Character Generation techniques.

PART B: Display devices, Interactive devices, Data generating devices, Thick lines.

UNIT 2: POLYGONS AND 2D TRANSFORMATION


2D Transformations: Introduction, Basic transformations such as- Scaling, Rotation, Translation, Homogeneous coordinates for transformations, Other transformations like – Reflection, Shearing Transformations, Transformations about an arbitrary point, Inverse transformations. Numerical problems on transformation

PART B: Fence fill algorithm, Problems on 2D transformation.

UNIT 3: SEGMENTS AND CLIPPING

PART A: Segment: Introduction, Segment table, Segment Creation, Closing, Delete and renaming, Image transformation, Display structures used for segment.


PART B: Liang-Barsky algorithm, Cyrus Beck algorithm.

UNIT 4: 3D TRANSFORMATIONS AND PROJECTIONS

PART A: 3D Transformations: Introduction, 3D point representation, Left handed system, Right handed system, Basic 3D transformations- Scaling, Rotation, Translation, Matrix representation, Derivation of Rotation matrices along the main axis, Rotation about an arbitrary axis, Reflection transformation with respect to main axes.

Projection: Projection concept, parallel and perspective projections, Viewing parameters, 3D windowing and clipping.
PART B: Reflection transformation with respect to planes, Problems based on 3D transformations

Unit 5: Hidden Surfaces, Curves and Fractals (7+1 Hrs)


Curve generation, Curve continuity, Interpolation, LeGrange interpolating algorithms, Spline curve representation, B Spline Curves, Bezier Curves, Fractals, Hilbert curve, Triadic Koch Curve, Fractal lines.

Introduction to light, Light Illumination models (Diffuse, Ambient, Specular), Point source illumination, Shading Algorithms (Phong, Gourad), Color, RGB Color Model.

PART B: Scan line algorithm for Depth Comparison, Fractal surfaces, CMY and HSV color model

Text Books:

Reference Books:

Course Outcomes:
Upon completion of the course, the graduates will be able to …
1. Apply mathematics and computer programming to computer graphics applications and problem solutions.
2. Utilize algorithms to draw, fill and clip basic geometrical shapes.
3. Construct animation based demonstrating system using segments and transformations.
4. Systematically identify and solve technical and aesthetic problems of transformations and projections.
5. Interpret the models of illumination, colors and hidden surfaces for representation of interactive graphics systems.
6. Devise and frame new set of algorithmic principles for efficient solutions useful to computing community.
CS20105::PRINCIPLES OF PROGRAMMING LANGUAGES

Credits: 03
Teaching Scheme: - Theory 3 Hrs/Week

Unit 1: Programming language concepts (8+2 Hrs)
PART B: Compilers and Interpreters. Just In Time interpreters.

Unit 2: Object – Oriented Programming (Java-I) (8+2 Hrs)
PART B: Pointers in C++, Destructors in C++.

Unit 3: Object – Oriented Programming (Java-II) (8+1 Hrs)
B Exception Handling in C++, multiple inheritance in C++.

Unit 4: Functional Programming using SCHEME (8+2 Hrs)
PART A: Introduction to lambda calculus. The Scheme programming Language: Atoms, Lists, lambda expressions. Functions as first class objects. Control structures, Recursion and continuations, operations on objects, basic input output, Exceptions and conditions, lazy evaluation and streams.
B Haskell

Unit 5: Introduction to SCALA (8+2 Hrs)
with XML. Actors and concurrency. GUI programming. Combining SCALA and JAVA.

**Text Books:**


**Course Outcomes:**

Upon completion of the course, graduates will be able to -

1. Develop solutions to problems using various programming paradigms.
2. Create programs in each paradigm based on context.
3. Differentiate real world problems based on domains.
4. Apply programming constructs for effective program design.
5. Integrate complexity issues in designing solutions using programming constructs.
6. Show good team work while working on team assignments and mini projects involving programming languages.
CS21112 :: DATA COMMUNICATION

Credits: 02  
Teaching Scheme: - Theory 2 Hrs/Week

Prerequisites:

Unit 1 : Introduction to Electronic Communication (6+1 Hrs)
PART B : Survey of communication applications. Numerical based on Shannon-Hartley theorem, Bandwidth calculation.

Unit 2 : Modulation Techniques (6+1 Hrs)
PART B : FM vs. AM, FM vs. PM, AM vs. PM, Numerical based on AM, FM.

Unit 3 : Multiplexing and Communication (5+2 Hrs)
PART A: Introduction, FDM, TDM, WDM, CDMA, Frequency modulation principles, sideband and modulation index. Pulse code modulation, Delta modulation, Adaptive delta modulation, Differential PCM, PAM,
B. Parallel transmission, serial transmission.

Unit 4 : Digital Modulations (5+1 Hrs)
PART B : Cellular Telephone System, Examples on Encoding, decoding.

Unit 5 : Transmission and Propagation (6+1 Hrs)
PART B : Transmission media wired and wireless, Microwave antenna.

Text Books
Reference Books


Additional Reading


Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Enumerate basics of signals, multiplexing, modulation and transmission.
2. Apply modulation and multiplexing techniques to optimize the channel requirements.
3. Compute the bandwidth, throughput, channel efficiency for different multiplexing and modulation techniques.
4. Recommend encoding techniques for communication system.
5. Justify the modulation, encoding and multiplexing techniques for specified communication system.
6. Evaluate the performance of network using error detection and correction methods.
CS20114:: MICROPROCESSOR AND INTERFACING

Credits: 03  Teaching Scheme: - Theory 3 Hrs/Week
Prerequisites: Digital Electronics and Logic Design.

Unit 1 : 8086 Microprocessor (8+1 Hrs)
PART A: Introduction to 80x86 microprocessor, Internal Architecture, Generation of physical address, Minimum & Maximum Mode, Ready and Reset pin significance .study of 8086 supporting chips 8282(Latch), 8284(Clock Generator), 8286(Transreceiver), 8288(Bus Controller). Timing Diagram Read Write Machine Cycles, Real Mode, General Purpose Instructions.
PART B : Instruction Set

Unit 2 : Assembly Language Programming & Interrupt Structure (8+1 Hrs)
PART A: Address Translation, Addressing Modes, Introduction to Assembly Language Programming, Examples on Programming. Interrupt Structure , Interrupt service Routine, Interrupt Vector Table, Hardware and Software Interrupts, INTR ,NMI , Interrupt Response, Execution of an ISR, Priority of Interrupts.
B. Examples on Assembly Language Programming

Unit 3 : Interfacing with 8086 – I (8+1 Hrs)
8255 (Programmable peripheral interface 8255)-block diagram, control word, interfacing ADC (Successive Approximation Method) , DAC (R – 2R ladder Network.)
PART B : Interfacing of stepper motor, seven segment display, (8255)

Unit 4 : Interfacing with 8086 – II (8+1 Hrs)
PART A : 8253/8254 –(Programmable Interval timer/counter) block diagram, control word & interfacing, Mode0, Mode1, Mode3 of timer,
8251(USART): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming.
PART B : Programming of 8251, Programming of Timer Mode 1, Mode2, Mode 3, Mode4, Mode5

Unit 5 : DMA Controller and NDP Co processor (8+1 Hrs)
PART A : Concept of DMA, 8237 DMA Controller: Features, Block Diagram & programming detail. 8087(NDP) - Features, Block Diagram, Control & status registers, typical Instruction set & programming detail.
PART B : Programming of 8237 and 8087(NDP).

Text Books

Page 54 of 251
Reference Books


Additional Reading


Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Describe the Structure and Internal Architecture of Microprocessor and Microprocessor Peripherals.
2. Develop simple Programs.
3. Assess a Component Requirement to solve a Computing Solution.
4. Design Interconnects of Microprocessor Peripherals.
5. Validate design outputs using standards test equipments.
CS20114:: MICROPROCESSOR AND INTERFACING

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:

List of Contents

A TERM-WORK containing the record of the following:

Assignments:

1. Understanding of different Memory Models
2. List various assembler directives,
3. Understand concepts of editor, assembler, linker, loader.
4. 8086 assembly language programming, to understand the basic concepts of various functions(01,02,08,09,0A) of INT 21h
5. List various debugging commands.
6. Interface 8086 microprocessor with 4KB RAM in minimum mode. Apply memory banking. Draw memory address map and explain address decoding logic.
7. Interface 8086 microprocessor with 16KB ROM in maximum mode. Draw memory address map and explain address decoding logic.
8. Design specified time delay (delay time calculation).
9. Near, Far procedures (string example).
10. Use of string instructions
11. Study of Mother Board

Note: Students should perform vi & vii assignments on drawing sheet
**Vishwakarma Institute of Technology**  
**Issue 05 : Rev No. 1 : Dt. 30/03/15**

**Text Books**

**Reference Books**
CS21104:: MATHEMATICAL TRANSFORMS AND APPLICATIONS

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:: Syllabus covered unit-wise in theory lectures.

List of Contents

In this module students will work on problems to practice and apply methods introduced in the theory lectures. Discussions of problems in small groups is always encouraged and facilitated. Students are asked to submit weekly home work assignments and provide them immediate feedback and support materials.

Tutorial No. 1: Summary on higher order linear differential equations, solution of homogeneous and non homogeneous equations, complementary solution.

Tutorial No. 2: Summary on particular solution by method of variation by parameters Method of undetermined coefficients and problems solving.

Tutorial No. 3: Summary on Euler – Cauchy and Legendre Equation, simultaneous equations and problems solving.

Tutorial No. 4: Summary on Functions of complex variables, Differentiation of functions of complex variables, Analytic functions, Harmonic functions, Harmonic conjugate.

Tutorial No. 5: Summary on Integration of functions of complex variables, integration along a path, Cauchy’s theorem, Cauchy’s integral formula, Cauchy’s residue theorem and problems solving.

Tutorial No. 6: Power Series, Taylor Series, Laurent Series

Tutorial No. 7: Summary on Laplace transform, Laplace transform of standard functions, Properties of Laplace and problems solving.

Tutorial No. 8: Summary on Properties of Laplace transform, Laplace transform of Unit step function, Dirac Delta function, Periodic functions and problems solving.

Tutorial No. 9: Summary on Inverse Laplace transform, properties of inverse Laplace transform, solution of differential equations by Laplace transform method and problems solving.
Tutorial No. 10: Summary on Fourier series, Complex form of Fourier series, Fourier integral representation and problems solving.

Tutorial No. 11: Summary on Fourier transform, Sine transform, Cosine transform and corresponding inverse and problems solving.

Tutorial No. 12: Summary on Discrete Fourier Transform, properties and problems solving.


Text Books

Page 59 of 251
CS20314:: MICROPROCESSOR AND INTERFACING

Credits: 01
Teaching Scheme: - Laboratory  2 Hrs/Week

Prerequisites:

List of Practical

Group A: - (Any 6)

1. Study of 8086 Architecture and Execution of sample programs.

2. Write 8086 Assembly language program to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.

3. Write 8086 ALP to perform block transfer operation. (Don’t use string operations) Data bytes in a block stored in one array transfer to another array. Use debugger to show execution of program.

4. Write 8086 ALP to find and count negative number from the array of signed number stored in memory.

5. Write 8086 Assembly language program (ALP) to arrange the numbers stored in the array in ascending as well as descending order. Assume that the first location in the array hold the number of elements in the array and successive memory location will have actual array elements. Write a separate subroutine to arrange the numbers in ascending and descending order. Accept a key from the user.
   a. If user enters 0, Arrange in ascending
   b. If user enters 1, Arrange in descending

6. Write 8086 Alp to convert 2_digit HEX number into equivalent BCD number.

7. Write 8086 ALP to convert 2_digit BCD number into equivalent HEX number.
Group B: - (Any 6)

1. Write 8086 Assembly language program (ALP) for following operations on the string entered by the user.
   a. Concatenation of two strings
   b. Find number of words, characters

2. Write 8086 ALP to convert an analog signal in the range of 0V to 5V to its corresponding digital signal using successive approximation ADC.

3. Write 8086 ALP to interface DAC & generate following waveforms on oscilloscope. Comment on types of DAC’s and write detailed specifications of the DAC used
   i) Square wave -- Variable Duty Cycle & frequency.
   ii) Stair case wave
   iii) Triangular wave

4. Write 8086 ALP to rotate a stepper motor for
   a. one clockwise rotation
   b. one anti clockwise rotation

   Write routines to accelerate and de-accelerate the motor
   Modify your program to rotate stepper motor for given angle and given direction.

5. Write 8086 ALP to program 8253 in Mode 0. Generate a square wave with a pulse of 10 mS.

6. Write 8086 ALP to initialize 8279 & to display characters in right entry mode. Provide also the facility to display “SECOMP”.
   a. Character in left entry mode
   b. Rolling Display
   c. Flashing Display

7. Perform an experiment to establish communication between two USART’s. Initialize USART-A in asynchronous transmitter mode and interface USART-B by initializing it in asynchronous receiver mode.

Note: - Students should perform any 6 assignments from group A and any 6 assignments from group B.

Text Books
Reference Books

CS20305: PRINCIPLES OF PROGRAMMING LANGUAGES

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C

List of Practical

1. Write a C++ program to implement the concept of objects, classes, constructors, destructors, inheritance.

2. Write a C++ program to implement the concept of static polymorphism (function and operator overloading.)

3. Write a C++ program to implement the concepts of virtual function (dynamic polymorphism) and friend functions

4. Write a C++ program to use the concept of generic programming (generic functions and generic classes)

5. Write a JAVA program to implement the concept of class, constructor, instance variable & class variable.

6. Write a JAVA program to implement the concept of inheritance, interface & package.

7. Write a Java program to implement the concepts of static polymorphism (function overloading) and dynamic polymorphism (using function overriding)

8. Write a JAVA program to use multithreading

9. Write a JAVA program for file handling.
10. Write a Java program with Graphical User Interface as a
    a) Stand alone desktop application
    b) Applet

11. Develop a web page using HTML, CSS and JavaScript

12. Familiarity with other programming languages
    a) Create a family hierarchy using PROLOG
    b) Write a .m file for matrix applications using MATLAB
Text Books

Reference Books

Additional Reading
CS20310::COMPUTER GRAPHICS

Credits: 01
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C

List of Practical

1. Write a Program to implement DDA and Bresenham Line drawing algorithm.
2. Write a Program to implement Bresenham’s Circle drawing algorithm.
3. Write a Program to implement Polygon fill algorithm.
4. Write a Program to implement Scaling, rotation and translation of a 2D object.
5. Write a program to achieve animation by using segmentation.
6. Write a Program to implement Cohen Sutherland line clipping algorithm.
7. Write a Program to implement Polygon clipping algorithm.
8. Write a Program to implement Scaling, reflection about planes and axes of a 3D object.
9. Write a Program to draw a Koch curve, fractal line and surface.

Text Books

Reference Books
CS27402: MINIPROJECT

Credits: 02

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -

7. Recognize essential & dominant area of technology for achievable artifacts over rapid period of time.
8. Acquire rapid application development cycle involving prototyping to learn adequate technological environments.
9. Concisely formulate specific problem in drafted specification format.
10. Devise data dictionaries and solution design with sufficient details.
11. Demonstrate the crafted solutions to user community with a lean learning curve.
12. Validate newer dimension of extendable and scalable nature of the problem solution crafting.
CS24302:: ASP.NET

Credits: 01  
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C#.NET or VB.NET

List of Practical

1. Design simple web application using ASP.NET.
2. Design web application with different validations.
3. Design on line database application.
4. Design data report application.
5. Design web application for uploading files on web.
6. Design AJAX application.
7. Design localized web application.
8. Design WPF browser application.
10. Deployment and publishing web sites.
11. Mini project.

Text Books

Reference Books

Course Outcomes:
Upon completion of the course, graduates will be able to -

The students should be able to
1. Analyze .NET framework, Common Language Runtime (CLR), garbage collection, and assemblies, forms, collections, constructs, delegates, events and exception handling.
2. Create data-driven web applications using the .NET Framework and ADO.NET.
3. Create web applications with rich UI and bug free experience using the ASP.NET standard Validation controls.
4. Evaluate problems and alternative web solutions using ASP.Net in a wide variety of business and organisational contexts.

Obtain hands-on on .Net Technologies to acquire responsible position in government and industry sectors.
CS24304:: Python

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C programming.

List of Practical

1. Syntax basics, Arithmetic/String Operations, Input/Output
2. Control Flow constructs: If-else, Relational and Logical Operators
3. Iteration: While loop, For loop
4. Collections: Lists, Tuples
5. Collections: Sets, Dictionary
6. Functions and Modules: sys, math, time
7. File Handling: Data streams, Access modes, Read/Write/Seek
8. Exception handling: hierarchy, raise, assert
9. OOP: Classes, Objects
10. GUI programming: TkInter

Text Books


Reference Books


Course Outcomes:

Upon completion of the course, graduates will be able to -
1. Develop functional, reliable and user friendly Python programs for given problem statement and constraints.

2. Correlate between the concepts of object oriented programming and the corresponding Python data structures while implementing programs using object oriented paradigm.

3. Judge a Python program in terms of correctness, space and time complexity and usability.

4. Adapt the existing solutions for familiar problems according to the needs/constraints of the similar problems.

5. Demonstrate competence through active participation in broader forums such communities supporting open source Python projects.

6. Utilize the problem solving and programming skills learned through the course for tackling relevant pressing issues in public and private sectors.
CS24307:: Ruby Programming

Credits: 01
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: C programming.

List of Practical

Note: Specific exercises will be given in class.

1. Introduction to Ruby. Ruby resources. Implement a simple Ruby program to print Hello World. Understand standard types.

2. Demonstrate the use of control structures.

3. Implement a Ruby program to demonstrate arrays and hashing.

4. Implement a Ruby program for Classes and Objects. Demonstrate class constructors and methods, inheritance, access control.

5. Implement a Ruby program to demonstrate containers, blocks and iterators.

6. Implement a Ruby program to demonstrate regular expressions.

7. Implement a Ruby program to demonstrate exceptions and Input Output

8. Implement threads in Ruby.

9. Implement a website using Scaffold.

10. Implement a dynamic website using RAILS. Use MySQL as the backend.

11. Implement a game in Ruby. Also implement a simple strategy for the computer to play the game.

12. Mini Project: As assigned. This will involve implementing some application in MVC using RAILS.

Text Books

1. Dave Thomas, Chad Fowler and Andy Hunt Programming Ruby 1.9 & 2.0 (4th
Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Integrate functional specifications into an outline of a solution in RUBY.
2. Develop pragmatic programming practices in RUBY.
3. Identify good verification and validation techniques for testing code.
4. Propose RUBY patterns for new and unfamiliar problems.
5. Build a useful body of programs in RUBY for the wider developer community.
6. Utilize the programming skills for problem solving in public and private sectors.

References:

2. [http://guides.rubyonrails.org/v3.2.9/index.html](http://guides.rubyonrails.org/v3.2.9/index.html)
MODULE V
<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs./week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lectures</td>
<td>Tutorial</td>
<td>HA</td>
</tr>
<tr>
<td>CS30101</td>
<td>Operating Systems</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS30116</td>
<td>Computer Networks</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS30105</td>
<td>Theory of Computation</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS30117</td>
<td>Human Computer Interaction</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS30109</td>
<td>Web Technologies</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td><strong>14</strong></td>
<td></td>
<td><strong>5</strong></td>
</tr>
</tbody>
</table>

**Laboratory Courses**

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs./week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Practical</td>
<td>ISE</td>
<td>ESE</td>
</tr>
<tr>
<td>CS30303</td>
<td>Operating Systems</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS30316</td>
<td>Computer Networks</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS30309</td>
<td>Web Technologies</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS37401</td>
<td>Mini Project (T.Y. Semester I) Irrespective of Module</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD1</td>
<td>Institute Level Elective</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS30401</td>
<td>Comprehensive Viva Voce</td>
<td>Based on Courses S1, S2,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS37301</td>
<td>Seminar (T.Y. Semester I)</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Irrespective of Module

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
<th>Total Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS37302</td>
<td>PROJECT STAGE 1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(T.Y. Semester II)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Irrespective of Module</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL**

<table>
<thead>
<tr>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

### Abbreviations Used

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>Home Assignment</td>
</tr>
<tr>
<td>TA</td>
<td>Teacher Assessment</td>
</tr>
<tr>
<td>ISE</td>
<td>In Sem. Examination</td>
</tr>
<tr>
<td>ESE</td>
<td>End Sem. Examination</td>
</tr>
<tr>
<td>CA</td>
<td>Continuous Assessment</td>
</tr>
<tr>
<td>*</td>
<td>Alternate Week Laboratory</td>
</tr>
<tr>
<td>^</td>
<td>Detail Enclosed Separately</td>
</tr>
</tbody>
</table>

**TOTAL CREDITS**

<table>
<thead>
<tr>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
</tr>
</tbody>
</table>
CS30101: OPERATING SYSTEMS

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures and Algorithms, Computer Organization.

Unit 1: Introduction to OS (8+1 Hrs)

PART A: Introduction to OS: What is OS, Interaction of OS and hardware, Goals of OS. Basic functions of OS, OS Services, System Calls.
Types of OS: Batch, Multiprogramming, Time sharing, Parallel, Distributed & Real-time OS.
Structures of OS: Monolithic, Layered, Ringed, Virtualization-Virtual Machines, Hypervisor, Exokernels, Client-server model, Microkernels.
Shell: Linux commands and shells, shell programming, AWK programming.
Introduction to Mobile OS: Architecture & Overview of Android OS.

PART B: Overview of Linux and Windows 2000 architecture

Unit 2: Process Management (8+1 Hrs)

Threads: Multithreading models, Thread implementations – user level and kernel level threads.
Symmetric Multiprocessing.
Concurrency: Issues with concurrency, Principles of Concurrency


Unit 3: Scheduling and Deadlock (8+1 Hrs)

Scheduling Algorithms: FCFS, SJF, RR, Virtual Round Robin, Priority
Multiprocessor Scheduling: Granularity, Design Issues, Process Scheduling

PART B: Thread Scheduling, Real Time Scheduling.

Unit 4: Memory Management (8+1 Hrs)

PART A: Memory Management concepts: Memory Management requirements, Memory Partitioning: Fixed, Dynamic Partitioning, Buddy Systems. Placement Strategies: First Fit, Best Fit, and Worst Fit, Fragmentation, Swapping, Paging,
Segmentation, Address translation.

**Virtual Memory**: Concepts, VM with Paging, Page Table Structure, Inverted Page Table, Translation Lookaside Buffer, VM with Segmentation.

**OS policies for Virtual Memory**: Fetch, Placement, Replacement, Resident Set management, Cleaning Policy, Load Control.


**PART B**: VM with combined paging and segmentation, Working Set Model.

**Unit 5**: I/O and File Management (8+1 Hrs)

**PART A**: I/O management: I/O Devices - Types, Characteristics of Serial and Parallel devices, OS design issues for I/O management, I/O Buffering.

**Disk Scheduling**: FCFS, SCAN, C-SCAN, SSTF.


**File System**: Structure, Implementation, Memory mapped files, Special Purpose File Systems

**Case study**: Process Management, Concurrency, Scheduling, Memory Management, I/O Management, File Management(VFS) in LINUX Shell and Command Programming, AWK Programming.

**PART B**: Organization of I/O functions, Disk Caches.

**Text Books**


**Reference Books**


**Additional Reading**

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Utilize mechanisms and strategies of an Operating System to solve real world problems which have similar requirements as the various core functionalities of an Operating System.
2. Explain the issues involved in design and implementation of operating systems in various contexts.
3. Analyze the tradeoffs inherent in operating system design.
4. Identify the objectives and functions of modern operating systems.
5. Examine the functions of a contemporary operating system with respect to convenience, efficiency and the ability to evolve.
CS30116:: COMPUTER NETWORKS

Credits: 03  Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Communication.

Unit 1: Introduction to Computer Networks and Logical Link Control (8+1 Hrs)
PART B: Point-to-Point Protocol (PPP), MPLS, Bridges, Gateways, Network Cables

Unit 2: Medium Access Control (8+1 Hrs)
PART A: Channel allocation: Static and Dynamic allocation, Multiple Access Protocols: Pure ALOHA, Slotted ALOHA, CSMA, WDMA, Ethernet: Cabling, MAC Sub-layer protocol: DIX and IEEE 802.3 Frame Formats, Collision Detection, Binary Exponential Back-off Algorithm, Switched Ethernet, Fast Ethernet, Wireless 802.11a/b/g/n LANS, MACA, Broadband wireless: 802.16
PART B: Gigabit Ethernet, Layer-II Switch and Bluetooth

Unit 3: Network Layer (9+2 Hrs)
PART B: Broadcast and Multicast routing, Routing for mobile hosts, IGMP, Mobile IP, VLAN

Unit 4: Transport Layer (8+1 Hrs)

PART B: Real Time Streaming Protocol RTSP, RTP, RTCP

Unit 5: Application Layer (7+1 Hrs)
PART A: Domain Name System (DNS), Naming and Address Schemes, DNS servers, E-mail: MIME, SMTP and POP3. Remote login, File Transfer Protocol (FTP), SNMP, DHCP and BOOTP. CDN, Working of Bit Torrent, Cloud computing: Architectures and working principle.

Text Books

Reference Books
3. Additional Reading

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Interpret topological network architectures and essential components to design it.
2. Estimate reliability issues based on error control, flow control and pipelining by using bandwidth, latency, throughput and efficiency.
3. Uniformly demonstrate LAN behavior utilizing network architecture, protocols, and network components.
4. Design client server based applications using sockets.
5. Demonstrate data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols.
CS30105:: THEORY OF COMPUTATION

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures.

Unit 1: Automata Theory


PART B: FA with output: Moore and Mealy machine.

Unit 2: Regular Expressions (RE) and Languages

PART A: Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen's Theorem, Regular expression to DFA, DFA to Regular expression, Non Regular Languages, Pumping Lemma for regular Languages, Myhill-Nerode theorem, Closure properties of Regular Languages, Applications of RE: Regular expressions in Unix, GREP utilities of Unix, Lexical analysis and finding patterns in text.

PART B: Decision properties of Regular Languages.

Unit 3: Context Free Grammars (CFG) and Push Down (11+1 Hrs) Automata (PDA)

PART A: Context Free Grammars: Definition, Examples, Derivation, Languages of Grammar, Derivation trees, Ambiguity in Grammar, Ambiguous and Unambiguous CFG, Inherent ambiguity, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure properties of CFLs, Decision Properties of CFLs (Emptiness, Finiteness and Membership), Chomsky Hierarchy. Pumping lemma for CFLs

Push Down Automata: Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof).

PART B: Regular grammars, left linear and right linear regular grammars, regular grammar and finite automata.

Unit 4: Introduction to Turing Machines

PART A: Turing Machines: Basic model, definition and representation, Instantaneous Description, Language acceptance by TM.

Robustness of Turing Machine model and equivalence with various variants: Two-
way/One-way infinite tape TM, multi-tape TM, non-deterministic TM, TM as enumerator.
Recursive and Recursively Enumerable languages and their closure properties.

PART B : Comparison between Finite Automata, Push Down Automata, and Turing Machines.

Unit 5 : Introduction to Undecidability (6+1 Hrs)
PART A : Universal Turing Machines, Church-Turing Thesis and intuitive notion of Algorithm.
Introduction to countable and uncountable sets (countability of set of natural numbers, integers, rationals. Uncountability of set of real numbers, points in plane), Encoding for Turing machines and countability of set of all Turing machines. Existence of Turing unrecognizable languages. Undecidability of Halting problem, Post Correspondence Problem. Example of a Turing unrecognizable language. Decision properties of R, RE languages and Rice’s theorem.

PART B: Hilbert’s tenth problem, undecidability of tiling problem

Text Books

Reference Books

Additional Reading

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. To infer the applicability of various automata theoretic models for recognizing formal languages.
2. To discriminate the expressive powers of various automata theoretic and formal language theoretic computational models.
3. To illustrate significance of non determinism pertaining to expressive powers of various automata theoretic models.
4. To comprehend general purpose powers and computability issues related to
state machines and grammars.

5. To explain the relevance of Church-Turing thesis, and the computational equivalence of Turing machine model with the general purpose computers.

6. To grasp the theoretical limit of computation (independent of software or hardware used) via the concept of undecidability.
CS30117:: HUMAN COMPUTER INTERACTION

Credits: 03          Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1 : Introduction (7+1 Hrs)
PART B: Identification of Application Category and Related Features for Selected Product / System.

Unit 2 : Principles, Models & Guidelines (9+2 Hrs)
PART B: Task / Error Analysis for Selected Product / System.

Unit 3 : Design Process and Interaction Styles (8+2 Hrs)
PART B: UI Design for Selected Product/System.

Unit 4 : Evaluation Techniques and Interface Categories (8+2 Hrs)
PART B: Usability Evaluation of Selected Product/System.

Unit 5 : Documentation and Groupware (8+1 Hrs)

PART B: Documentation Design for Selected Product/System.

Text Books

Additional Reading

Course Outcomes:
Upon completion of the course, graduates will be able to:

1. Identify human factors and usability issues related with computing applications
2. Differentiate computing applications into categories based on human factors
3. Design a user interface by applying suitable design principles, models and usability guidelines
4. Integrate ethno-cultural and accessibility computing aspects into the user interface design.
5. Display the impact of usability evaluation and testing in computing applications
6. Follow required processes and standards while designing user interfaces
CS30109 :: WEB TECHNOLOGIES

Credits: 02  
Teaching Scheme: 2 Hrs/Week

Prerequisites: Principles of Programming Languages.

Unit 1: Introduction to Web Technologies  
(6+1 Hrs)
PART A: World wide web, what we mean by "web technologies", Evolution of web: front content to social to semantic (Web 1.0, 2.0, 3.0, web as a platform) Introduction to Client side technologies: HTML, CSS, JavaScript, AJAX.
HTML: HTML Elements, Attributes, HTML formatting Tags for skeleton, text, links, lists, images, styles, Tables, Frames, Introduction to CSS, JavaScript: Data Types and Variables, Operators and expressions, Arrays, Loops, conditions, JavaScript Objects, Forms & Reg Expressions, Events, JavaScript Browser Functions.
PART B: Applications of JavaScript, Validations Using JavaScript, Creating Menu using CSS.

Unit 2: JQuery and Data formats and representations – I  
(5+1 Hrs)
HTML DOM: Predefined Objects, Object Hierarchy, Accessing Objects, Event Handlers, Node: Create, Add, Insert, Clone, Remove and Replace.
PART B: Creating Animations and Slider using JQuery, Referring DOM using JavaScript.

Unit 3: Data formats and representations – II  
(5+1 Hrs)
PART A: XML: Role of XML, Prolog, Elements, Attributes, Namespace XML DTD: Purpose of DTD, Using DTD in XML, Element Type Declaration, Attribute Declaration XSD: Limitations of DTD, Schema Elements, Element Definition, Schema Schema Validation, Built in Data Types. The Java API for XML parsing (JAXP), X-Path, Overview of XSLT.
PART B: X-query.

Unit 4: Server Side Technologies  
(6+1 Hrs)
JSP: JSP lifecycle, Directives, Comments, Expressions, Scriptlets, Declaration, Scope of JSP Objects, Standard Actions, Introduction to JavaBeans, Calling JavaBean from JSP Page. Client side security: Cookies security policy, Server side security tools e.g. Web Application Firewalls (WAFs) and fuzzers.
PART B: Session and Cookies management in Servlet and JSP.
Unit 5 : Distribution of data  
(5+1 Hrs)


PART B: AJAX: AJAX Benefits, Basic Idea, Asynchronous Communication, AJAX Processing Steps,

Text Books
4. “XML and Related Technologies” by Atul Kahate, PEARSON Education.

Reference Books

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Design static and dynamic web pages using HTML, PHP and XML Technologies.
2. Apply the effects of CSS, Javascript, JQuery and XSLT in Web design.
4. Evaluate problems and alternative web solutions using Web Technologies in a wide variety of business and organizational contexts.
5. Create data-driven web applications using JSP and Servlet.
6. Incorporate best practices for building applications with Struts.
CS30101:: OPERATING SYSTEMS

Credits: 01  
Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:: Data Structures and Algorithms, Computer Organization.

List of Contents

A TERM-WORK containing the record of the following:

1. Execution of Advance Unix commands.

2. Write a shell program to sort an array of numbers using any sort method.

3. Execution of AWK related commands.

4. Implement the solution for Reader-Writer problem using Threads and Semaphores/Mutex.

5. Implement the solution for Producer-Consumer (Bounded Buffer) problem using Threads and Semaphore/Mutex .

6. Implement the solution for Dining-Philosopher problem using Threads and Semaphore.

7. Implementation of resource allocation graph (RAG).

8. Implement the solution for Banker’s Algorithm for deadlock avoidance.

9. Draw the Gantt charts and compute the finish time, turnaround time and waiting time for the following algorithms:
   a. First come First serve
   b. Shortest Job First (Preemptive and Non-Preemptive)
   c. Priority (Preemptive and Non-Preemptive)
   d. Round Robin

10. Calculate the number of page faults for a reference string for the following page replacement algorithms:
    a. Optimal
    b. FIFO
    c. LRU
11. Calculate the total distance traversed by the disk arm to satisfy the pending requests for the following disk scheduling algorithms:
   a. FCFS
   b. SSTF
   c. SCAN
   d. C-SCAN

Text Books

Reference Books

Additional Reading:
CS30117:: HUMAN COMPUTER INTERACTION

Credits: 01  
Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:

List of Contents

A TERM-WORK containing the record of the following:

1. Design user persona for the users of selected product/system.
2. Perform GOMS analysis for selected product/system.
3. Conduct a contextual inquiry for selected product/system.
4. Design an interface prototype for selected product/system.
5. Evaluate an interface using usability testing / evaluation technique.

Text Books

Reference Books

Additional Reading
CS30303:: OPERATING SYSTEMS

Credits: 01
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Structures and Algorithms, Computer Organization

List of Practical
Part A:

1. Implementation of a multiprogramming operating system:
   a. Stage I:
      i. CPU/ Machine Simulation
      ii. Supervisor Call through interrupt
   b. Stage II:
      i. Paging
      ii. Error Handling
      iii. Interrupt Generation and Servicing
      iv. Process Data Structure
   c. Stage III:
      i. Multiprogramming
      ii. Virtual Memory
      iii. Process Scheduling and Synchronization
      iv. Inter-Process Communication
      v. I/O Handling, Spooling and Buffering

Text Books

Reference Books

Additional Reading
CS30316:: Computer Networks

Credits: 01  
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Communication

List of Practical

1. Set up a small network of 2 to 4 computers using Hub/Switch. It includes installation of LAN Cards, Preparation of Cables, Assigning IP addresses and sharing C drive.

2. File Transfer using PC To PC Communication.


4. Studying Linux and Windows network commands. [ ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap, etc]

5. Program for calculating CRC using Modulo-2 and Polynomial methods.

6. Simulate the sliding window protocols Go Back N and Selective Repeat.

7. File Transfer between two computers using TCP sockets.

8. Multiuser chat application using UDP sockets.

9. To create TCP/IP packet using standard TCP/IP include files and send it to other machine

10. Program to find active and passive ports on nearby host using sockets.


12. Installing and configuring DHCP server for Linux/Windows.

Text Books


Page 93 of 251
Reference Books

Additional Reading
Vishwakarma Institute of Technology  
CS30309::WEB TECHNOLOGIES  

Credits: 01  
Teaching Scheme: - Lab 2 Hrs/Week

Prerequisites: Principles of Programming Languages

List of Tutorials

1. Design a Simple web page using HTML.
2. Design an Attractive web page using CSS.
3. Design a web page using HTML and implement validations using Java Script and give effects using JQuery.
4. Design an XML page and there DTD and validate it using Java API for XML parsing.
5. Design an XML page and there XSD and validate it using Java API for XML parsing.
6. Design web page using a Java Servlet to read and write data to database using JDBC.
7. Design an interactive web page using JSP.
8. Design an interactive web page using PHP.

Text Books


Reference Books

CS37401::MINI PROJECT

Credits: 02

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes
Upon completion of the course, graduates will be able to -
1. Recognize essential & dominant area of technology for achievable artifacts over rapid period of time.
2. Acquire rapid application development cycle involving prototyping to learn adequate technological environments.
3. Concisely formulate specific problem in drafted specification format.
4. Devise data dictionaries and solution design with sufficient details.
5. Demonstrate the crafted solutions to user community with a lean learning curve.
6. Validate newer dimension of extendable and scalable nature of the problem solution crafting.
CS37301::SEMINAR

Credits: 02  
Teaching Scheme: - Lab 2 Hrs/Week

Seminar is a course requirement wherein under the guidance of a faculty member a student is expected to do an in depth study in a specialized area by doing literature survey, understanding different aspects of the problem and arriving at a status report in that area. Students are expected to choose a topic in CSE based on current trends or industry practices. While doing a seminar, the student is expected to learn investigation methodologies, study relevant research papers, correlate work of various authors/researchers critically, study concepts, techniques, prevailing results etc., analyze it and present a seminar report. Evaluation will be based on relevance of topic, understanding of the problem, literature Survey, presentation, communication skills, answering queries and reporting or documenting procedure.

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Scope and visibly identify technological trade off in computer engineering leading to significant topics.
2. Conduct a thorough literature survey of identify technical topic.
3. Present technical topic in written form with technical report or document
4. Communicate effectively technical topic in verbal form with suitable demonstration
5. Access real world problem scenarios in computer engineering.
6. Demonstrate skills and competences with an awareness of technical standardization.
CS37302::PROJECT STAGE I

Credits: 02

Aim
This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. The Project Work can lead to:

a. Transform existing Ideas into conceptual models.
b. Transform conceptual models into determinable models.
c. Use determinable models to obtain system specifications.
d. Select optimum specifications and create physical models.
e. Apply the results from physical models to create real target systems.

Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student’s own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.
3. The project must be based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting , Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
6. The Project Work will be assessed jointly by a panel of examiners. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.
7. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate’s performance based on
presentation skills, questions based on the Project Work, understanding of the Project, analysis and design performed for the project.

8. The Student Project Groups are expected to work on the recommendations given by the panel of examiners.

Assessment Scheme

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Content</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Concept</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>System Requirement Specification</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>System Analysis</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>System Design Block Diagram</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Presentation of the Project Work</td>
<td>10</td>
</tr>
</tbody>
</table>

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Identify Real World Problems
2. Apply Computing Solutions to Real World Problems
3. Construct a Solution Model to Real World Problem
4. Select Design Pattern to Best approach the Solution.
5. Lay Down rules to Minimise Adverse Impact of Design Implementation
6. Adapt to changing Technological and Human resource advances.

Note:
The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members.
The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.

Following is the list of recommended domains for Project Work:

<table>
<thead>
<tr>
<th>Computer Networks</th>
<th>Image Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Systems</td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td>Network Security</td>
<td>Expert Systems</td>
</tr>
<tr>
<td>Digital Signal Processing</td>
<td>Object Oriented Systems</td>
</tr>
</tbody>
</table>

Page 99 of 251
<table>
<thead>
<tr>
<th>Course</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Programming</td>
<td>Modeling and Design</td>
</tr>
<tr>
<td>Real Time Systems</td>
<td>System Testing</td>
</tr>
<tr>
<td>Embedded systems</td>
<td>Storage Management</td>
</tr>
<tr>
<td>Cluster Computing</td>
<td>Client-Server Computing</td>
</tr>
<tr>
<td>Mobile &amp; Wireless Communications</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>Multimedia Systems</td>
<td>Protocol Engineering</td>
</tr>
</tbody>
</table>
## Course Structure

### Theory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs. / week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lectures</td>
<td>Tutorial</td>
<td>HA</td>
</tr>
<tr>
<td>CS30102</td>
<td>Software Engineering</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CS30106</td>
<td>Database Management Systems</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CS30118</td>
<td>Advanced Data Structures and Algorithms</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CS31119</td>
<td>Object Oriented Modeling and Design</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CS30114</td>
<td>Systems Programming</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>14</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Laboratory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs. / week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Practicals</td>
<td>ISE</td>
<td>ESE</td>
</tr>
<tr>
<td>CS30314</td>
<td>System programming</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS30306</td>
<td>Database Management Systems</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS31319</td>
<td>Object Oriented Modeling and Design</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS37402</td>
<td>Mini Project (T.Y. Semester I) Irrespective of Module</td>
<td>2</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>CS37301</td>
<td>Seminar(T.Y. Semester I) Irrespective of Module</td>
<td>4</td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>PD</td>
<td>Institute Level Elective</td>
<td>Credits</td>
<td>Module</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------</td>
<td>---------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>CS30402</td>
<td>Comprehensive Viva Voce Based on Courses S2, S4</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CS37302</td>
<td>PROJECT STAGE 1 (T.Y. Semester II) Irrespective of Module</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>16</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

**Abbreviations Used**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>Home Assignment</td>
</tr>
<tr>
<td>TA</td>
<td>Teacher Assessment</td>
</tr>
<tr>
<td>ISE</td>
<td>In Sem. Examination</td>
</tr>
<tr>
<td>ESE</td>
<td>End Sem. Examination</td>
</tr>
<tr>
<td>CA</td>
<td>Continuous Assessment</td>
</tr>
<tr>
<td>*</td>
<td>Alternate Week Laboratory</td>
</tr>
<tr>
<td>^</td>
<td>Detail Enclosed Separately</td>
</tr>
</tbody>
</table>

**TOTAL CREDITS**

26
CS 30102: Software Engineering

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures

Unit 1: Software Engineering Paradigms  (8 Hrs)

Unit 2: Requirement Engineering  (8 Hrs)
Part B: Requirements Verification and Validation, Requirement Maturity, Technical Reviews

Unit 3: System Analysis and Design Foundations  (8 Hrs)

Unit 4: System Architecture Determination  (8 Hrs)

Part B: Architecture Analysis Techniques, Zachman Framework, Architecture Assessment

Unit 5: Project Management Principles (8 Hrs)


Part B: Classic Mistakes, Complex Systems, Critical Systems, Software Safety

Text Books:

Reference Books:

Course Outcomes:

Upon completion of the course, graduates will be able to –
1. Identify the process models required to construct software in order to quench stakeholder needs and requirements.
2. Interpret the problem scope associated with real world problems.
3. Compose software artifacts with conformation to stated requirements.
4. Evaluate the criteria required to balance overall problem solution pair.
5. Demonstrate synergistic and cohesive team work that justifies solution realizations.
6. Build realistic solution assembled by either technological availability or through creativity.
CS30106:: DATABASE MANAGEMENT SYSTEMS

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data structures

Unit 1 : Introduction and Data Models  
(9+2 Hrs)


PART B : Spreadsheet Model, Codd's Twelve Rules for Relational DBMS, Life Cycle of a Relational Database

Unit 2 : Database Design Theory  
(7+2 Hrs)

PART A : Normalization: Need, Functional Dependency, Inference Rules, FD Closure, Minimal Cover, Decomposition Properties, Normal Forms (upto BCNF), Multi-valued Dependency (4NF), Relational Synthesis Algorithm

PART B : Join and Inclusion Dependency, 5NF, DKNF, Trade - off

Unit 3 : Query Languages  
(6+2 Hrs)

PART A : Formal Relational Query Languages: Relational Algebra, Tuple Relational Calculus; SQL: DDL, DML, Select Queries, Join Queries, Subqueries; PL/SQL: Procedure, Function, Trigger; Query by Example;

PART B : Domain Relational Calculus, DCL-Security and Authorization, Date-Timestamp, String and Numerical Functions, Mapping of Relational Algebra to SQL

Unit 4 : Storage and Querying  
(9+2 Hrs)


PART B : SAN, Files with Sparse / Dense Index; Query Processing: Sort Operation, Impact of Indices on Query Performance;

Unit 5 : Transaction Management and Emerging Trends  
(9+2 Hrs)


Page 106 of 251
PART A: Transaction: ACID Properties, Concurrency Control Protocols: Lock-based, Multiple Granularity, Multiversion Scheme; Failure and Recovery; NoSQL: RDBMS vs NoSQL, BASE properties, NoSQL Categories; NewSQL; Emerging Trends: Distributed Databases, Distributed Data Storage, Distributed Query Processing; Parallel Databases, Architectures, Speedup and Scaleup, Decomposition, Data Replication; Time Series Databases, Spatial and Geographic Databases;

PART B: Design of Core DBMS Functions, Timestamp based Concurrency Control Protocol, ARIES Recovery Technique, Personal Databases

Text Books

Reference Books
3. "Getting Started with NoSQL: Your guide to the world and technology of NoSQL", by Gaurav Vaish

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Develop a database system using relational database query languages, PL/SQL and
NoSQL.
2. Construct refined logical database model with consideration of data semantics and dependency.
3. Design data models to enforce data requirements and operational constraints of an organization.
4. Describe techniques used by a DBMS for data storage, access and query processing.
5. Describe various database system architectures and their functionalities.
6. Formulate alternative queries for given data requirement considering the query evaluation plan.
CS30118 :: ADVANCED DATA STRUCTURES AND ALGORITHMS

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures

Unit 1 : Advanced searching and retrieval data structures  (8+1 Hrs)
PART B: Alternate hash functions (mid-square, folding, digit analysis), LCS problem

Unit 2 : Priority Queues and Advanced Heap structures  (8+1 Hrs)
PART B: Comparative study of different priority queue implementations using – binary heaps, leftist trees, binomial heaps, Fibonacci heaps with respect to the following operations – insert, delete, find-min, extract-min, decrease-key, meld

Unit 3 : Overview of Time Complexity analysis, Divide and Conquer  (8+1 Hrs)
PART A: Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations) and time complexity, Overview of searching, sorting algorithms (binary search, insertion sort, heap sort, bubble sort), lower bound for comparison based sorting. Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity and to prove correctness of algorithms.
PART B: Finding Maximum and Minimum, Convex Hull problem, Master’s Theorem’s and its cases

Unit 4 : Greedy Strategy and Dynamic Programming  (8+1 Hrs)
PART B: String Editing Problem, Scheduling problem, Optimal Storage Problem.

Unit 5 : NP-Theory (8+1 Hrs)


PART B: Decision Vs Search versions of problems in class NP, Coupon Collector Problem, Bin Packing.

Text Books


Reference Books


Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Interpret the design and implementation of advanced data structures.
2. Analyze the time and space complexity algorithms using asymptotic notations.
3. Apply appropriate algorithm design technique and data structures to solve problems.
4. Grasp the notion of intractability, NP-Complete and NP-hard problems.
5. Appraise the role of randomization and approximation in computation.
6. Make intelligent decisions about data structures and algorithmic techniques in the context of practical software problems, choosing from existing data structures and algorithms or designing their own when necessary.
CS30114:: SYSTEMS PROGRAMMING

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:
- Data Structures
- Computer Organization
- Microprocessors

Unit 1: Introduction to System Programming (6+1 Hrs)
Assemblers: Elements of Assembly language programming, Simple assembler scheme, Structure of an assembler, Design of single and two pass assembler.
Macro Processors: Macro Definition and call, Macro expansion, Nested Macro Calls, Advanced Macro Facilities, design of Macro Preprocessor. Booting Procedure for DOS & Windows.
PART B: RISC machines, Machine dependent and machine independent Assembler features.

Unit 2: Compilers, Loaders and Linkers (5+1 Hrs)
PART A: Compilers: Introduction to Compiler phases, Introduction to cross compiler, Features of machine dependent and independent compilers, Overview of types of compilers, Interpreters.
Linkers: Relocation and linking concepts, Static and dynamic linker, subroutine linkages.
Loaders: Introduction to Loader, Loader Schemes: Compile and go, General Loader Scheme, Absolute loaders, relocating loaders, direct linking loaders.
PART B: Instruction description, Pseudo operations, Instruction Mapping, MSDOS Linker, Sun OS linker.

Unit 3: Essential concepts of Systems programming for Linux as Open Source OS.
PART A: Introduction and essential concepts of LINUX system programming: System Programming, APIs and ABIs, standards, Program segments/sections; The ELF Format, Linking and loading, Linux dynamic libraries (shared objects), Multitasking and paging, Address translation, Memory Protection, Comparison with Windows.

PART B: Dynamic linking, API compatibility, Dynamically linked libraries, Overall architecture and limitations.

Unit 4: Encoding, Decoding and Device drivers (6+1 Hrs)
PART B: Library Description for IA-32/Intel64.

Unit 5: TSR Programming (5+1 Hrs)

Text Books

Reference Books

Course Outcomes
Upon completion of the course, graduates will be able to -

1. Develop different system software like Macroprocessor, Assembler, Compiler, Linker and Loader.
2. Discriminate among different System software and their functionalities.
3. Design Device Drivers, TSR programs and DLL for real world applications.
4. Solve critical problems related to Encoding, Decoding and Instruction set.
5. Conforms to use proper data structures for system programming.
6. Follow methods and techniques for implementing system-level programs.
CS31119 : Object Oriented Modeling and Design

Credits: 03  
Teaching Scheme: - Theory  3 Hrs/Week

Prerequisites: Data Structures

Unit I : Introduction to Modeling  
( 8 Hrs )
Part B: Methodology: Coad-Yordon, Responsibility-Driven Design, OPM, Catalysis, Aspect-Oriented Modeling

Unit 2 : Model Driven Development  
( 8 Hrs )
Part B: Domain-Specific Modeling: Fundamentals and Architecture, MDA Applications

Unit 3 : Behavior Specification  
( 8 Hrs )
Part A: Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Using Use Cases, The Domain Perspective, Data Dictionary: Finding the Objects, Responsibilities, Collaborators, and Attributes, CRC Cards, Class Models and Use Case Models, Judging the Domain Model, Capturing system behavior in use cases
Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization
Part B: Study of other Behavioral Diagrams: Communication Diagram, Interaction Overview Diagrams, Timing Diagrams

Unit 4 : Design Specification  
( 8 Hrs )
Part A: Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, Categorizing classes:
entity, boundary and control, Modeling associations and collections, Preserving referential integrity, Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors
Forward Engineering and Reverse Engineering Concepts
Part B: Application of UML in Real Time and Embedded System, Application of UML in Web Engineering, UML Profiles for other technology disciplines

Unit 5: Design Patterns (8 Hrs)
Part B: Antipatterns, Applications of Design Patterns, Archetype Patterns

Text Books:

Reference Books:

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Determine the unfamiliarity of the problem frames in order to envisage conceptual nomenclature.
2. Break down system functionalities into realizable customer-centric and developer-centric situations.
3. Narrate design specifications in terms of industries-specific practices such as methodology–driven engineering.
4. Propose multi-faceted defendable solutions with overt-behavior demonstrating team-skills.
5. Initiate new problem issues and compatible solution aspects with the help of design pattern.
6. Automatically devise solution terminologies reducing the potential of cost and performance impedance.
CS 30102: Software Engineering

Credits: 01  
Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites: Data Structures

List of Contents

A TERM-WORK containing the record of the following:

1. To study Software Process Models and identify their applicability to various categories of projects.

2. To understand Requirement Elicitation Techniques and recognize types of requirements while preparing System Requirement Specification.


4. To develop all level Data Flow diagrams for the target system indicating problem partitions and solution structure.

5. To apply design principles with relevant architecture style and structure the solution accordingly.

6. To prepare estimation for the System Development using Function Point technique.

Text Books:

Reference Books:
CS30118:: Advanced Data Structures And Algorithms

Credits: 01

Prerequisites: Data Structures

Teaching Scheme: - Tutorial 1 Hr/Week

List of Contents

1. Example to indicate the limitations of static hashing and how it is addressed using dynamic hashing.
2. Implement an application that uses bloom filter.
3. Improve the performance of Dijkstra’s shortest path algorithm using Fibonacci heaps.
4. Implement a binomial heap and compare its amortized complexity with binary heap.
5. Implement an application that makes use of a prefix tree (trie) – address book, spell checker, auto completion etc.
6. Implement a solution for LCS problem (Longest Common Subsequence) using suffix trees.
7. Numerical Problems based on
   a. Mathematical Induction,
   b. Solving recurrence relations,
   c. Proof by contradiction.
8. Time complexity comparison of matrix multiplication by using
   b. Strassens Matrix multiplication.
10. Implementation of travelling sales person problem using Dynamic programming technique.
11. Implementation of OBST or 0/1 knapsack problem using dynamic programming technique.
12. Implementation and time complexity comparison of Randomized and non-randomized version of algorithm for MIN_CUT problem by using different test cases.
13. Implementation and time complexity comparison of Randomized and non-randomized version of algorithm for Quick sort problem by using different test cases.
15. Introduction to Linear programming and study of (Simplex Algorithm, Ellipsoid Algorithm, Interior Point Algorithm).
16. Study of parallel algorithms and parallel programming techniques.

Text Books


CS30314:: SYSTEMS PROGRAMMING

Credits: 01  Teaching Scheme: - Lab 2 Hrs/Week

List of Assignments:

1. Expanding the Simple Macros with Generating different Parameter Tables and MDT.

2. Expanding the Nested Macros with Generating different Parameter Tables and MDT.

3. Design and implementation of 1 pass and 2 Pass assemblers with generating different data structure for it.

4. Design and implementation of an Editor: Design of a Line or Screen Editor using C Language.

5. Symbol table generation for input *.c file.


7. Simulation of linkers.

8. Simulation of loaders.

9. Understanding the design for DLL on Linux shared library.

10. Use of different debugger tools.

11. Printer controller in device drivers.

12. Write a TSR program in 8086 ALP to implement Real Time Clock (RTC). Read the Real Time from CMOS chip by suitable INT and FUNCTION and display the RTC at the bottom right corner on the screen. Access the video RAM directly in your routine.

13. Write a TSR program in 8086 ALP to implement Screen Saver. Screen Saver should get activated if the keyboard is idle for 7 seconds. Access the video RAM directly in your routine.

14. Write a TSR program in 8086 ALP to handle the “Divide by zero” interrupt. Test your program with a small code, which causes the divide by zero interrupt.
15. Write a TSR program in ‘C’ that would change the color of the screen every 10 seconds.

Note: It is expected that student must perform at least 2 assignments from assignment number 12 to 15.

Text Books

Reference Books
CS30306: DATABASE MANAGEMENT SYSTEMS

Credits: 01  
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

1. Choose a database system you propose to work on throughout the course. Perform requirements analysis in detail for design of the database. Design an entity-relationship (ER) data model for the selected database system.

2. Convert above ER model to relational model, semi structured data model. List functional dependencies. Normalize these relations up to 3NF/BCNF.

3. Consider a different database system. List functional dependencies [Include complex business logic.] Apply bottom-up approach using Relational Synthesis Algorithm for design of relational model for the chosen system. Verify decomposition properties.

4. Create tables with appropriate constraints for the relational schema. Create views, indices, and sequence. Alter the schema by adding/removing columns and constraints. Write DML queries.

5. Execute ‘SELECT’ queries using order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.

6. Write equijoin, non equijoin, self join and outer join queries. Write queries containing single row / multiple row / correlated subqueries using operators like =, in, any, all, exists etc. Write DML queries containing subqueries. Study a set of query processing strategies.

7. Write meaningful stored procedures in PL/SQL. Make use of cursors and different arguments. Write useful stored functions to perform complex computation. Write row level and statement level triggers in PL/SQL.

8. Implement a small database application for the above system using suitable front end and back end tool. Create a transaction by embedding SQL into an application program. Generate different useful reports.

9. Implementation of a small database using NoSQL and/or New SQL database system.
Text Books


Reference Books


3. "Getting Started with NoSQL: Your guide to the world and technology of NoSQL", by Gaurav Vaish

CS31319: Object Oriented Modeling and Design

Credits: 01  
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data Structures

List of Practical

1. To narrate Requirement Definition Document for the target system with following three areas:
   a. Problem Identification
   b. Problem Definition
   c. Problem Statement


2. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases.

The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
   a. Actors (External Users)
   b. Transactions (Use Cases)
   c. Event responses related to transactions with external agents.
   d. Detection of System boundaries indicating scope of system.

3. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
   a. Discrete, distinguishable entities (class).
   b. Events (Individual stimulus from one object to another).
   c. Conditional events and relationship representation.

4. To depict the state transition with the life history of objects of a given class model. The model should depict:
   a. Possible ways the object can respond to events from other objects.
   b. Determine of start, end, and transition states.

5. To depict the dynamic behavior using detailed Activity diagram.

6. To develop logical static structure of target system with Software Class diagram.

To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
b. Identification of objects and their purpose.
c. Roles / responsibilities entities that determine system behavior.

7. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.

8. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. To implement the system according to specification with confirmation to design patterns.

Text Books:

Reference Books:
CS37402::MINI PROJECT

Credits: 02

The Student has to select a project in group based on a topic of interest from any of the subjects offered in current Semester. Periodically the implementation will be evaluated by the guide.

Evaluation is done in two stages. In the first review the internal Guide evaluates the project against 40% of the implementation of work. At the end of semester each group will be evaluated by externally Guide from Industry based on their Presentation, completeness of Project implementation and report artifact.

Course Outcomes

Upon completion of the course, graduates will be able to -
1. Recognize essential & dominant area of technology for achievable artifacts over rapid period of time.
2. Acquire rapid application development cycle involving prototyping to learn adequate technological environments.
3. Concisely formulate specific problem in drafted specification format.
4. Devise data dictionaries and solution design with sufficient details.
5. Demonstrate the crafted solutions to user community with a lean learning curve.
6. Validate newer dimension of extendable and scalable nature of the problem solution crafting.
CS37301::SEMINAR

Credits: 02

Teaching Scheme: - Lab 2 Hrs/Week

Students are expected to choose a topic in CSE based on current trends or industry practices. They should prepare present a report and present it after having fully understood the concepts. Evaluation will be based on relevance of topic, literature Survey, presentation, communication skills, answering queries and reporting or documenting procedure.

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Scope and visibly identify technological trade off in computer engineering leading to significant topics.
2. Conduct a thorough literature survey of identify technical topic.
3. Present technical topic in written form with technical report or document
4. Communicate effectively technical topic in verbal form with suitable demonstration
5. Access real world problem scenarios in computer engineering.
6. Demonstrate skills and competences with an awareness of technical standardization.
## Course Structure

**Branch – Information Technology**  
**Year -** THIRD Year of Engineering  
**Semester -** 2  
**Academic Year – 2015-2016**  
**Pattern – f11**

### Theory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs./week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lectures</td>
<td>Tutorial</td>
<td>HA</td>
</tr>
<tr>
<td>CS40115</td>
<td>Data Acquisition Systems</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS40114</td>
<td>Business Intelligence and Analytics</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS42105</td>
<td>Mobile Computing</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS42127</td>
<td>Cloud Computing</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42103</td>
<td>Parallel Computing on GPU</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42101</td>
<td>Advanced Computer Graphics</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42131</td>
<td>Enterprise Systems</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42135</td>
<td>Digital Image Processing</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42112</td>
<td>Distributed System</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td>14</td>
<td>2</td>
<td>14</td>
</tr>
</tbody>
</table>

### Laboratory Courses

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme (Hrs./week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Practicals</td>
<td>ISE</td>
</tr>
<tr>
<td>CS40115</td>
<td>Data Acquisition Systems</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS40114</td>
<td>Business Intelligence and Analytics</td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CS42105</td>
<td>Mobile Computing</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS42127</td>
<td>Cloud Computing</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS42103</td>
<td>Parallel Computing on GPU</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS42101</td>
<td>Advanced Computer Graphics</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS42131</td>
<td>Enterprise Systems</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS42135</td>
<td>Digital Image Processing</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CS42112</td>
<td>Distributed System</td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Credits</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------</td>
<td>---------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>CS40315</td>
<td>Data Acquisition Systems</td>
<td>2</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Elective I</td>
<td>2</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>CS47303</td>
<td>PROJECT STAGE 2 (B.Tech Semester I)</td>
<td>8</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Irrespective of Module</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>12</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Abbreviations Used**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>Home Assignment</td>
</tr>
<tr>
<td>TA</td>
<td>Teacher Assessment</td>
</tr>
<tr>
<td>ISE</td>
<td>In Sem. Examination</td>
</tr>
<tr>
<td>ESE</td>
<td>End Sem. Examination</td>
</tr>
<tr>
<td>CA</td>
<td>Continuous Assessment</td>
</tr>
<tr>
<td>*</td>
<td>Alternate Week Labotatory</td>
</tr>
<tr>
<td>^</td>
<td>Detail Enclosed Separately</td>
</tr>
</tbody>
</table>

**TOTAL CREDITS**

26
CS40115::DATA ACQUISITION SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Unit 1 : Systems and Measurements

PART A : Introduction to data acquisition system(DAS), Block Diagram of DAS, Multichannel DAS, Generalized measurement system, Characteristics of measuring devices, accuracy, precision, errors, linearity, hysteresis, threshold, repeatability, reliability calibration., Basic concepts of feedback control system:- open loop and closed loop.

PART B : Applications of open loop and closed loop systems

Unit 2 : Sensors and Signal Conditioning circuits

PART A : Introduction to the sensors, Detectable phenomenon, choosing of sensors, Types of sensors: Temperature sensor (LM35), Light sensor(photodiode, optocoupler), Distance and range sensor(IR), Accelerometer sensor, Gyroscope sensor, Touch screen sensor. Signal conditioning circuits: Analog and Digital, Opamp in signal conditioning circuits as amplifier.

PART B : Case study for RTD, LVDT and Thermocouple sensors

Unit 3 : ARM Microcontroller

PART A : Family Architecture, Block diagram, Features, Data / Programme Memory, Reg set, Reg Bank, Special Function Registers, Data Memory, Programme Memory, Interrupt Structure , Timer Prog ,Serial Port Prog , Misc Features.

PART B: Memory I/O Design & Interfacing, Timer Calculation

Unit 4 : I/O interfacing & Bus Systems


PART B : USB, Centronics ,

Unit 5 : Programmable Logic Controls

PART A : PLC : Design, PLC Operation: I/O scan mode, execution mode, PLC Software Functions: Timer, accumulated timer, counters. Elements of Ladder Diagrams (limit,
pressure, level, thermal, mechanical switch) and examples based on ladder diagrams. Data Loggers.

PART B : SCADA

Text Books:

Reference Books :

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Define the characteristics of measuring devices for an instrument.
2. Identify the functioning of measuring devices in an industrial process.
3. Design elements of signal conditioning circuit that are necessary for sensors.
4. Describe the structural units of Microcontroller.
5. Interconnect devices using communication buses.
6. Develop programs for the process control systems.
CS40113:: BUSINESS INTELLIGENCE AND ANALYTICS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Database Management Systems.

Unit 1 : Introduction (8+1 Hrs)

Data Warehousing & Multi-Dimensional Modeling: What is a data warehouse, need for a data warehouse, four characteristics of a data warehouse, components/architecture of a data warehouse, data marts, building a data warehouse. OLTP vs OLAP.

Multidimensional Modeling: ER Modeling vs Multi-Dimensional Modeling, Data cubes, OLAP operations, lattice of cuboids, multi-dimensional schemas (star, snowflake, galaxy), facts, dimensions, concept hierarchies, types of dimensions, measures

Data Integration: different types of digital data and their sources, ETL (extract-transform-load)

Enterprise Reporting: Metrics, Measurement, Measures, KPIs, Dashboards, Reports, Scorecards

PART B: Examples of business analyses – funnel analysis, distribution channel analysis and performance analysis; levels of decision making (strategic, tactical and operational BI); types of facts; types of measures; different OLAP architectures

Unit 2: Descriptive Analytics - I (8+1 Hrs)
PART A: Data Preprocessing: mechanisms of data collection and challenges involved therein. Typical preprocessing operations: combining values into one, handling incomplete or incorrect data, handling missing values, recoding values, subsetting, sorting, transforming scale, determining percentiles, data manipulation, removing noise, removing inconsistencies, transformations, standardizing, normalizing - min-max normalization, z-score standardization, rules of standardizing data.

Data visualization: role of visualization in analytics, different techniques for visualizing data based on the nature of data and what kind of insights need to be drawn

Descriptive Statistics: role of statistics in analytics, types of data (scales of measurement - NOIR), data distributions, measures of variability (range, quartile, five number summary, variance, std dev, coeff of variation), analyzing distributions, Chebychev’s Inequality, measures of shape (skewness, kurtosis), measures of association (covariance, correlation), outliers

PART B: Data quality and its various aspects, measures of central tendency (arithmetic mean, geometric mean, harmonic mean, median, mode)
Unit 3: Descriptive Analytics - II

PART A: Inferential Statistics: Role of probability in analytics. Need for sampling, generating samples, sampling and non-sampling error. Sampling Distribution of Mean, Central Limit Theorem, Standard Error.

Estimation: Point and Interval Estimates, Confidence Intervals, level of confidence, sample size.

Hypothesis Testing: basic concepts, Errors in hypothesis testing, Power of test, Level of significance, p-value, general procedure for hypothesis testing. Parametric tests – z test, t test, chi-square test. Hypothesis testing of means: two tailed and one-tailed tests. Chi-square test for independence and goodness of fit. Hypothesis testing for comparing two related samples. Limitations of hypothesis testing. Picking up the right test for a given scenario.

PART B: Common probability distributions and their characteristics, parametric vs non-parametric tests, ANOVA

Unit 4: Predictive Analytics

PART A: Similarity Measures: Design of recommender systems - user based and item based collaborative filtering

Modelling: Data Modelling Basics, Logic driven modeling, data driven modeling

Regression Analysis: Correlation and regression, Simple Linear Regression Model, Least Squares Method. Making Data Models more flexible, making data models more selective, dealing with Categorical variables, Interpretation of regression coefficients, fine tuning data models (assessing the fit, model fitting), Coefficient of determination, Significance tests, Residual analysis, Prediction intervals. Model evaluation techniques. Assumptions of regression analysis.

Introduction to Time Series Analysis and Forecasting: Time series patterns, forecast accuracy, moving averages and exponential smoothing, casual models, using regression analysis for forecasting, determining best forecast model to use, ARIMA models.

PART B: Basic what-if spreadsheet models, different types of regression and which is used when, modeling non-linear relationships, heteroscedasticity, multicollinearity

Unit 5: Prescriptive Analytics & Emerging Trends in BI&A

PART A: Optimization Analytics: Overview of simulation and risk analysis, Linear Optimization Models (linear programming), Integer Linear Optimization models (integer programming), Non-linear optimization models (portfolio theory), Monte Carlo Simulation, Decision Analysis

BigData Analytics: What is Big Data, sources of BigData, MapReduce, Hadoop, statistical analysis of big data, visualizing bigdata

Familiarity with the following areas: multi variate analytics, text analytics, web analytics, social media analytics, in-memory analytics – row vs columnar databases, in-memory databases.

PART B: NoSQL databases for unstructured data and its role in analytics, analytics in the cloud, embedded BI, mobile analytics, mobile BI, Data Stream Analytics, Customer Centricity, Patient Centricity, IoT, Self-Service BI, Fog networks
Text Books
6. “Business Analytics” by James R Evans, Pearson

Reference Books
1. “Business Intelligence for Dummies”
5. “Information Dashboard Design: The effective visual communication of data”, Stephen Few, O’Reilly

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Construct an end-to-end data warehousing solution for business intelligence involving various data sources, ETL, multi-dimensional modeling, OLAP, reporting and analytics
2. Evaluate various data processing algorithms in their applicability to different problems
3. Display the process of converting data into a user defined format required for particular analysis
4. Utilize statistical tools in deriving insights from data
5. Describe various techniques for descriptive, predictive and prescriptive analytics
6. Apply various techniques to solve real-world data analysis problems
CS42105:: MOBILE COMPUTING

Credits: 03  Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:
- Communication Engineering
- Computer Network

Unit 1: Cellular Network (8+1 Hrs)
PART B: Signal and signal propagation, IEEE 802.11standard, 802.11a,b,g, 802.15,

Unit 2: GSM Communication (8+1 Hrs)
PART A: System Architecture: GSM Radio subsystem, Interfaces, Network and switching subsystem, Operation subsystem. GSM channels, GSM protocol architecture, Location tracking and call setup, Security, Data services N/W signaling, GSM mobility management, Administration and maintenance.
Handoff- Initialization of handoff, Delaying handoff, Forcing handoff, Power different handoff. Mobile assisted handoff, Intersystem handoff.
PART B: Survey of GSM network, Hard, Soft Handoff.

Unit 3: GSM Bearer Services (9+1 Hrs)
PART A: SMS architecture protocol, Hierarchy, VOIP services for mobile networks.
WAP: model and architecture, Gateway, protocol stack.
Telecommunication system: GPRS, wireless in local loop, DECT, EDGE, UMTS, Paging systems, CDPD.
PART B: Wireless application environment, Bluetooth.

Unit 4: Mobile Network and Transport layer (8+1 Hrs)
Mobile Transport layer: Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast retransmit/recovery, Transmission/time-out freezing, Selective retransmission, Transaction-oriented TCP.
PART B: TCP over 2.5/3G wireless networks.

Unit 5: Mobile Databases (7+1 Hrs)
PART A: Database hoarding, Data caching, Data cache and web cache maintenance in mobile environments, Client-Server computing and adaptation, Query processing, Data recovery process, Issues relating to quality of service, Digital audio broadcasting: DAB System, DAB objects, Object transfer protocol, DVB: DVB system.

PART B: Mobile Billing

Text Books

Reference Books

Course Outcomes:
Upon completion of the course, the students will be able to:
1. Describe the functional specification of 2G and 3G Cellular Network Standards.
2. Compute performance parameters for designing the Cellular Network.
3. Propose Telecommunication system to be deployed to fulfill bandwidth capacity planning.
4. Justify the Mobile Network performance parameters and design decisions.
5. Predict the requirements of next generation mobile network.
6. Design Mobile Application to solve a real world problem.
CS42127::CLOUD COMPUTING

Credits: 03
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Networks, Operating Systems, Web Technologies

Unit 1: Introduction (6+1 Hrs)


PART B: Study of top 5 cloud providers and top 5 applications on cloud

Unit 2: Architecture and IaaS (8+1 Hrs)
PART A: Cloud Computing Architecture: Cloud computing stack - Comparison with traditional computing architecture (client/server), Services provided at various levels, How Cloud Computing Works, Role of Networks in Cloud computing, protocols used, Role of Web services. Service Models (XaaS) - Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS). Deployment Models, Public cloud, Private cloud, Hybrid cloud, Community cloud


PART B: Study top 5 open source tools for building private cloud along with pros/cons

Unit 3: PaaS and SaaS (7+1 Hrs)
PART A: Platform as a Service (PaaS): Introduction to PaaS - What is PaaS, Service Oriented Architecture (SOA), Cloud Platform and Management - computation, storage

Software as a Service (SaaS): Introduction to SaaS, Web services, Web 2.0, Web OS, Case Study on SaaS

PART B: Innovative applications of cloud computing

Unit 4: Cloud Management and Security (10+1 Hrs)

Page 137 of 251
Processing


PART B: Study the cloud deployment of a big enterprise

Unit 5: Cloud Solutions (9+1 Hrs)
PART A: Case study on Open Source and Commercial Clouds – Amazon EC2, Google Compute Engine, Microsoft Azure, Cloudfoundry, OpenStack
PART B: How the CDNs (Content Delivery Networks) make use of the cloud?

Text Books
3. Cloud Security by Ronald Krutz and Russell Dean Vines, Wiley-India

Reference Books
2. Cloud Security & Privacy by Tim Malhar, S.Kamarasswammy, S.Latif (SPD.O’REILLY)
4. Cloud Computing Bible by Barrie Sosinsky, Wiley India.
5. Cloud Computing, Michael Miller, Que Publishing

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Describe the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
2. Explain the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Collaboratively research and write a paper on the state of the art (and open problems) in cloud computing.
4. Identify problems, and explain, analyze, and evaluate various cloud computing solutions.
5. Choose the appropriate technologies, algorithms, and approaches for the related issues.
6. Display new ideas and innovations in cloud computing.
CS42103:: PARALLEL COMPUTING ON GPU

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Parallel programming basics (8+1 Hrs)
PART A: Parallel programming definition, motivation, Types and levels of parallelism, Different grains of parallelism, data dependence graph, data parallelism, functional parallelism, Flynn’s classification of multi-processors, Motivation for heterogeneous programming, Definition of thread and process, programming parallel computers - extend a compiler, extend a sequential programming language, add a parallel programming layer, create a parallel language.
PART B: multiprocessor and multicomputer systems, interconnection networks

Unit 2: Introduction to GPU, architecture and parallel algorithms (8+1 Hrs)
PART A: Introduction to GPU computing, motivation, Modern GPU architecture case study: NVIDIA Fermi Tesla C2050/Kepler K20, GPU memories- global, shared, texture memory and their properties and uses, roles of CPU and GPU in parallel computing, GPU computing domain areas and success. Parallel algorithm design. Speedup and scalability.
PART B: parallel algorithm design for data clustering, theory of locality of reference

Unit 3: Compute Unified Device Architecture (CUDA) (8+1 Hrs)
PART A: CUDA Architecture, CUDA programming model, execution model, thread organization: Concept of grid, block and thread, thread index generation, warp; memory model: Introduction to global, shared, local memories, usage of cache, texture cache, constant memory, memory banks and bank conflicts, memory coalescing, CUDA structure and API details. CUDA example programs (Vector dot product, Vector-Matrix multiplication and etc).
PART B: atomic operations in CUDA

Unit 4: Problem solving using GPUs (8+1 Hrs)
PART A: Single vss double precision, solving problems that involves Vectors, Matrices, Binomial coefficients, Bernstein coefficients and etc. Instructor will choose the problems from several domains.
PART B: Study problems given by instructor

Unit 5: Parallel reduction and Tools
PART A: Reduction operation using prefix sum example. Performance issues in algorithms- deciding parallelization of a part of algorithm and selecting the highest parallelism, Need of profilers, Introduction to CUDA Tools: MemCheck and Visual Profiler.

PART B: Memory leaks and associated problems

Text Books
2. CUDA by Example: An Introduction to General-Purpose GPU Programming by Jason Sanders and Edward Kandrot
3. Parallel Programming in C with MPI and OpenMP by Michael J. Quinn, Tata McGraw-Hill Edition

References:
2. www.tutorials on introduction to parallel computing
3. Other references suggested by instructor

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Analyze the real problem for exploiting maximum parallelism on GPU architecture
2. Solve the complex problems using GPUs
3. Compare serial and parallel executions.
4. Code and optimize the parallel programs on GPU using CUDA.
5. Apply parallel computing methods to research oriented problems.
CS42101:: ADVANCED COMPUTER GRAPHICS

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Graphics

Unit 1: OpenGL (8+1 Hrs)
PART A: OpenGL over windows, SDK, Extensions, GLUT, GLU, OpenGL primitives, Programming language: Blending, 3D viewing(camera analogy), Lighting model, Culling, Fog, Texture mapping. 
PART B: OpenGL over Linux, pBuffer rendering, Shadowing Techniques.

Unit 2: Graphics Hardware (8+1 Hrs)
PART A: Graphics card, Components of graphics card, General-Purpose computation, GPU, GPGPU, CUDA architecture, CUDA Programming: Type qualifiers, Execution configuration, vector types, functions, APIs. 
PART B: How graphics card works? GeForce 6800 series GPU architecture.

Unit 3: Advanced Rendering Techniques (8+1 Hrs)
PART A: Point based rendering, Mesh Simplification, Spatial partitioning, Solid Modeling, Subdivision surfaces: Catmull-Clark subdivision, Subdivision rules, Visibility Computation: culling types, cells and portals, hardware support. 
PART B: Splines, Tessellation, 3D viewing.

Unit 4: Photorealistic and Volume Rendering (8+1 Hrs)
PART B: Monte Carlo mathematical formulation, Marching cubes algorithm.

Unit 5: Texture Synthesis and Image Processing (8+1 Hrs)
PART A: Texture synthesis, Image processing: Digital image representation, Image data structures, Sampling and Quantization, Image enhancement in spatial domain. 
PART B: Image compression, Image synthesis.

Text Books

Reference Books
Page 141 of 251

Additional Reading

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Design algorithms/programs for 2D/3D graphics using OpenGL.
2. Discriminate data structures and algorithms of 3D modeling, visibility and rendering.
3. Describe graphics hardware for its functions and programming.
4. Distinguish algorithms of global illumination and 3D volume rendering.
5. Conform the use of image and texture processing for graphics.
6. Construct creative 2D/3D graphics models.
CS42131: Enterprise Systems

Credits: 03  Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Software Engineering, Database Management System

Unit 1: Business Process Management

Process Orchestrations: Control Flow Patterns, Workflow Nets, Business Process Modeling Notation
Process Choreographies: Motivation and Terminology, Development Phases, Process Choreography Design and Implementation, Service Interaction Patterns
Properties of Business Processes: Data Dependencies, Structural Soundness, Soundness, Relaxed Soundness, Weak Soundness, Lazy Soundness, Soundness Criteria Overview

Part B: Petri Nets, Event-driven Process Chains, Yet Another Workflow Language, Graph-Based Workflow Language

Unit 2: SOA Fundamentals

Service Contracts (Standardization and Design): Contracts principles, Types of Service Contract Standardization, Contracts and Service Design, Versioning, Technology/Development Tool Dependencies
Part B: XML; HTTP; SOAP

Unit 3: SOA Design Principles

Part A: Service Abstraction (Information Hiding and Meta Abstraction Types): Abstraction principles, Types of Meta Abstraction, Measuring Service Abstraction, Service Abstraction and Service Design, Risks Associated with Service Abstraction
Service Reusability (Commercial and Agnostic Design): Reuse Principle, Service Reuse in SOA, Service Reusability and Service Design

Service Autonomy (Processing Boundaries and Control): Autonomy Principle, Types of Service Autonomy, Measuring Service Autonomy, Service Contract Autonomy (services with normalized contracts) Autonomy and Service Design

Service Statelessness (State Management Deferral and Stateless Design): State Management, Measuring Service Statelessness, Statelessness and Service Design

SOA Delivery Strategies, Service-Oriented Analysis: Introduction, Service Modeling, Service-Oriented Design: Introduction, SOA Composition Guidelines), Service Design,

**Part B:** Importance of WSDL, SOAP, The use of registries via UDDI

---

**Unit 4: SOA Technology and Implementation**

**Part A:** Service Discoverability (Interpretability and Communication): Discoverability, Types of Discovery and Discoverability, Measuring Service Discoverability, Discoverability and Service

Service Composability (Composition Member Design and Complex Compositions): Composition, Composition Concepts and Terminology, Complex Service Composition, Measuring Service Composability and Composition Effectiveness Potential, Composition and Service Design, Service-Orientation and Object-Orientation, Mapping Service-Orientation Principles to Strategic Goals

**Part B:** SOA Platforms, SOA support in .NET and J2EE platforms

---

**Unit 5: Enterprise Architecture**


---

**Text Books:**


**Reference Books:**


Page 144 of 251

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. To model business requirements and business processes using BPMN 2.0 standard
2. To discover the set of services with composite services creation and designing services to facilitate integration and understand interrelationships among SOA, Web Services, OOD and an IT infrastructure
3. To focus Enterprise SOA and facilitate use of Enterprise Service Bus in an Enterprise SOA
4. To explore the concepts and technology for service orchestration and discuss the guidelines to integrate a Business Process Management Solution in an Enterprise SOA.
5. To practice the concepts, specifications, and technologies for service composition and explore the role of an enterprise service bus in an Enterprise SOA.
6. To understand case studies, lessons learned, and best practices for planning and implementing SOA projects
CS42135: Digital Image Processing

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Digital Image Fundamentals & Basics of Digital Image Processing  6+1 Hrs
PART B : RGB to Grey conversion, RGB to HSV conversion

Unit 2: Prepossessing  8+1 Hours
PART A : Introduction to Spatial Domain and Frequency Domain Processing, Convolution, Spatial Domain Image Enhancement Techniques: Denoising filters, Smoothing Operation, Sharpening Operation, Contrast enhancement (Histogram Equalization), Frequency Domain Processing: 2 dimensional Fourier transform of an image, filtering in Fourier domain
PART B : Frequency domain noise reduction

Unit 3: Segmentation  9+1 Hours
PART B : Line detection and corner detection

Unit 4: Image Processing  8+1 Hours
PART A : Morphological Operations: Dilation, Erosion, Opening, Closing, Connected Components, Object Recognition: Based on Decision-Theoretic Methods, Based on Structural Methods
PART B : Convex Hull, thinning and pruning operations.

Unit 5: Case Studies  9+1 Hours
PART A : License Plate Recognition, Traffic Sign Detection and Recognition, Image Color Effects
PART B : Study of Various Image Effects in popular mobile apps

Text Books

Reference Books

Page 146 of 251
Course Outcomes

1. Convert the image colour model.
2. Determine and implement required image enhancement techniques.
3. Apply various corrective geometric transforms on a distorted image.
4. Deploy faster algorithms for image domain conversions which ensures expected performance on variety of hardware architectures.
5. Implement an algorithmic solution using open source technologies such as OpenCV.
6. Develop image processing solution for social and personal security.
CS42125: Randomized and Approximation Algorithms

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:
Prior introduction to basic probability theory is helpful though it is not presumed. First course on algorithms.

Unit 1: Basic probability theory (8+1 Hrs)
PART A: Introduction to randomization in computation and some simple randomized algorithms. Basic discrete probability theory: basic counting, definition of probability, examples, independence of events, conditional probability, union bound, inclusion exclusion, Bayes’ rule, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, standard distributions (Bernoulli, Binomial, Geometric), coupon collector problem, birthday paradox, probabilistic recurrences. Uniform generation of combinatorial structures. Indicator random variables and their role in algorithm analysis.
PART B: Review Las Vegas and Monte-Carlo algorithms (with examples: randomized quick sort, Karger’s min-cut algorithm.)

Unit 2: Tail inequalities, randomized complexity classes (8+1 Hrs)
PART A: Tail Inequalities and applications: Moments and deviation, occupancy problem, Markov and Chebyshev inequalities and some applications, randomized selection, weak law of large numbers, stable marriage problem and principle of deferred decision, coupon collector problem and sharp threshold, Chernoff’s bound and some applications, set balancing.
Complexity classes: Basic complexity classes P, NP, RP, Co-RP, ZPP, BPP and their interrelations, probability amplification in RP and BPP, randomness and nonuniformity, Adleman’s theorem.
PART B: Yao’s min-max principle and lower bound for randomized computations.

Unit 3: Algebraic techniques (8+1 Hrs)
PART A: Polynomial identity testing, Schwartz-Zippel lemma and applications (with examples verifying matrix multiplication, testing equality of strings, perfect matching problem for bipartite graphs), Mulmuley-Vazirani-Vazirani isolation lemma and application to matching problem. Number theoretic algorithms (finding quadratic non-residues, primality testing), introduction to probabilistic methods.
PART B: Application of randomized algorithms in geometric problems (Convex hulls, half space intersection, Delaunay triangulations, diameter of point set).

Unit 4: Markov Chains and Random Walks (8+1 Hrs)
PART A: Markov chains: definition, representations, randomized algorithm for 2-SAT and 3-SAT, classifying states of Markov chains, Gambler’s ruin, stationary distributions. Random walks on undirected graphs, cover time, hitting time, commute time, graph connectivity, electrical networks, introduction to expander graphs.

PART B: Expanders and rapidly mixing random walks.

Unit 5: Approximation Algorithms (8+1 Hrs)
PART A: Introduction to approximation algorithms, NP-hard optimization problems, lower bounding OPT, example of set-cover (O(log n) factor approx-algorithm based on greedy strategy, layering), Shortest super-string problem, Knapsack and FPTAS algorithms. Linear programming based algorithms, LP relaxation, LP duality. LP rounding strategy and primal-dual schema, set-cover and some other examples using LP based techniques, maximum satisfiability.
PART B: Review Vertex cover problem, Traveling Salesman Problem (TSP), general TSP hard to approximate, metric TSP.

Advanced Topics:
(depending on the availability of time some of the following topics may be included)
Role of expander graphs in derandomization, hardness of approximation, statement of PCP theorem, approximation algorithms for classical problems on integer lattices.

Text Books
2. Approximation Algorithms by Vijay V. Vazirani (Springer)

Reference Books

Course Outcomes:

Upon completion of the course, graduates will be able to
1. To solve problems based on the basic discrete probability and combinatorics
2. To design Las-Vegas, Monte-Carlo randomized algorithms for various computational problems
3. To analyze time complexity and success probability of randomized algorithms using random variables.
4. To illustrate application of tail inequalities in tight estimation of the success
probability and the time complexity of randomized algorithms

5. To explain role of advanced algebraic techniques such as Schwartz-Zippel Lemma, Isolation Lemma, Markov chains and random walks on graphs in randomized algorithms design

6. To design approximation algorithms for NP-complete problems using suitable paradigm
CS42136: DISTRIBUTED COMPUTING

Credits: 03
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Operating Systems.

Unit 1:
Introduction

Introduction to Hadoop/MapReduce.
PART B: Conceptual design of a Distributed File System based on concepts learned in Unit I. Reading: Hadoop/MapReduce.

Unit 2: Communication

PART B: Case studies of existing RPC implementation. Design a RPC/RMI mechanism based on the case studies and concepts learned in class.

Unit 3: Synchronization

PART A: Time and Global States: Clock Synchronization, Logical Clocks, Scalar time, Vector time, Global State, Event Ordering.
Co-ordination: Election Algorithm: Bully Algorithm, Ring Algorithm, Mutual Exclusion: Requirements, Performance metrics, Centralized Approach, Lamport’s algorithm, Distributed Approach (Ricart and Agrawala)
Distributed Deadlock Algorithms for Avoidance, Prevention, and Detection: Classification of distributed deadlock detection algorithms, Centralized Approach, Hierarchical Approach, WFG Based Fully Distributed, Deadlock Recovery.
PART B: Comparative analysis of time synchronization/mutual exclusion/election algorithms implementations in well-established distributed systems.

Unit 4: Fault Tolerance

(8+1 Hrs)
Reliable Client Server Communication, Reliable Group Communication.

PART B: Identify the issues that can arise in your DFS prototype in case of various failures. Based on the concepts learned in Unit 4: design a fault tolerance mechanism for the DFS.
Develop test-case scenarios to assess the system's resilience.

Unit 5: Distributed Transaction and Distributed Shared Memory (8+1 Hrs)
PART A: Distributed Transaction: Transaction Model, Classification, Implementation, Concurrency Control: Serializability, 2 Phase Locking, Strict 2 PL, Distributed Commit: 2 Phase Commit, Recovery

PART B: Distributed File Systems: Data Intensive Computing, Google FS, BigTable

Text Books

Reference Books

Additional Reading

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify the basic principles, design requirements and challenges associated with implementing large-scale distributed systems.
2. Develop the solutions for Communication, Synchronization, Mutual exclusion and Deadlock handling in distributed computing.

3. Compare existing distributed systems in terms of their choice of algorithms and approaches.

4. To examine state-of-the-art distributed systems such as Google File System, Big Table, Data Intensive Computing.

5. Propose an optimal and cost-effective solution without compromising the security and reliability of the system.

6. Apply principles of distributed computing while building appropriate variations of existing solutions to meet the development contexts.
CS42119:: INFORMATION RETRIEVAL

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures, Database Management Systems

Unit 1: Information Retrieval Introduction and Models (9 Hrs)
PART A: Introduction, Definition, Objectives, Search and Browse Capabilities; A Formal Document Representation, Characterization of IR Models, Boolean Retrieval, Extended Boolean Models, Vector Space Model, Probabilistic Model, Measure of Relatedness, Term Weighting, Naive Bayes Text Classification, Document and Term Clustering, Flat and Hierarchical Clustering, Matrix Decomposition, Latent Semantic Indexing
PART B: Bayesian Model, Data Structures and Techniques for Ranking, Models for Browsing, Zipf's law, Heap's law

Unit 2: Query Processing and Retrieval Evaluation (8 Hrs)
PART A: Digital libraries, Morphological, Lexical Analysis, Stemming Algorithms, Thesaurus Construction, Ontology, Metadata, Query Languages, Similarity Measures and Ranking, Relevance Feedback, Query Expansion, Retrieval Performance, Evaluation Measures for Ranked and Unranked Results
PART B: Porter's Stemming Algorithm, Automatic Local/Global Analysis, Information Summarization and Visualization, Archiving and Preservation

Unit 3: Indexing and Searching (8 Hrs)
PART A: Automatic Indexing, Inverted Files, Structures Used, Signature Files, Compression, Partitioning, Tries, Suffix Trees and Suffix Arrays, Index Construction, Distributed Indexing, Index Compression, Sequential Searching, Pattern Matching, String Matching allowing Errors, Regular Expressions and Extended Patterns, Pattern Matching using Indices, Structural Queries
PART B: Fast Inversion (FAST-INV) Algorithm, Algorithms on PAT Tree, Faceted Search

Unit 4: Parallel, Distributed IR and Web Searching (8 Hrs)
PART A: Parallel IR, Index Construction, Distributed IR, Characterizing the Web, Search Engines, Browsing, Metasearchers, Searching using Hyperlinks, Crawling, Link Analysis, Architectures (Agents, Buses, Wrappers/Mediators)
PART B: Watermarking, PageRank Algorithm, HillTop Algorithm

Unit 5: Multimedia IR (7 Hrs)
PART A: Multimedia Data Modeling, Query Languages, A Generic Multimedia Indexing (GEMINI) Approach, One Dimensional Time Series, Two Dimensional Color Images, Automatic Feature Extraction, Operations on images, Motion detection, Object recognition, Automatic image annotation and retrieval, Audio, Graph, Video Retrieval
PART B: Hashing Algorithms, Image Features and Similarity Functions

Text Books

Reference Books
3. “Information Retrieval”, C. J. Van Rijsbergen, Information Retrieval Group, University of Glasgow, online at http://www.dcs.gla.ac.uk/Keith/Preface.html

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Describe various information retrieval system architectures and models
2. Validate retrieval performance of an information retrieval system
3. Construct various indexes using suitable techniques
4. Apply sequential search and pattern matching techniques
5. Illustrate working of parallel, distributed and multimedia information retrieval system
6. Explain various information retrieval algorithms and different types of queries
CS42113:: DIGITAL SIGNAL PROCESSING

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Introduction to signals and systems (8+1 Hrs)


PART B: Properties of LTI systems, parallel and cascade connection, Correlation of

Unit 2: Z and Fourier transforms (8+1 Hrs)


PART B: Symmetry properties of F. T, F. T. theorems: Linearity, time shifting, frequency shifting, time reversal, differentiation.

Unit 3: Frequency analysis of Signals and Systems (8+1 Hrs)


Sampling the F.T., Fourier representation of finite-duration sequences, The Discrete Fourier Transform, Properties of DFT: circular shift, duality, symmetry, Circular Convolution, Linear Convolution using DFT, Effective computation of DFT and FFT, DIT FFT, Overlap and save algorithm, Inverse DFT using FFT(DIF)

PART B: Properties of DFT: Linearity, DIF FFT, Goertzel Algorithm, Inverse DFT using FFT(DIT)

Unit 4: Design of Digital Filters (9+1 Hrs)
PART A: Concept of filtering, Ideal filters and approximations, specifications, IIR filter design from continuous time filters: Characteristics of Butterworth, Chebyshev approximations, impulse invariant and bilinear transformation techniques, Design examples, FIR filter design using windows: properties of commonly used windows, systems with Linear phase, Generalized Linear phase systems, Four Types of GLPS (TypeI), Design Examples, Design using Kaiser window, Comparison of IIR and FIR Filters.

PART B: Four Types of GLPS (Type II,III,IV), Examples on Filter Design (IIR & FIR)

Unit 5: Realization of Filters (7+1 Hrs)


PART B: Instruction set of ADSP 21XX series processor and some examples.

Text Books

Reference Books

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Convert analog signal into a digital signal without irreversible data loss using mathematical techniques.
2. Design faster algorithms for signal domain conversion which ensures expected operation on all architectures.
3. Design filters meeting the given specifications with the help of Matlab.
4. Design a stable system which requires minimum components to implement.
5. Analyse a signal in different mathematic domains to understand the signal characteristics.
6. Differentiate between various realizations techniques.
CS31314::DATA ACQUISITION SYSTEMS

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

A: Study of Opamp in signal conditioning circuits as amplifier:

1. Inverting Amplifier.
2. Adder and Subtractor.
3. Integrator and Differentiator.

B: Interfacing of Microcontroller for the following:

1. I/O devices with Key, Switch and LED
2. LCD device
3. RS232C with PC
4. ADC device
5. Temperature Sensor
6. Optocoupler
7. LDR
8. Accelerometer
9. Touch Sensor

C: Mini project based on interfacing of microcontroller

Text Books:


Reference Books:

CS42125:: RANDOMIZED AND APPROXIMATION ALGORITHMS

Credits: 01  
Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Prior introduction to basic probability theory is helpful though it is not presumed. First course on algorithms.

List of Contents

A TERM-WORK containing the record of the following:

1. Problem solving based on basic probability (random variables, mean, variance)
2. Problem solving based on basic probability (linearity of expectation, some applications, conditional probability)
3. Use of indicator random variables in algorithm analysis
4. Tail Inequalities and applications (Markov, Chebyshev)
5. Tail Inequalities and applications (Chernoff bound, variance of sum of co-related random variables)
6. Random walks, rapid mixing and some applications like sampling points from convex bodies
7. Schwartz-Zippel Lemma and applications
8. Complexity of perfect matching problem
9. Randomization in Number theoretic and geometric algorithms
10. Approximation algorithms, lower bounding OPT
11. Linear programming and LP-duality, LP based approximation algorithms
12. PCP theorem and hardness of approximation
Text Books
2. Approximation Algorithms by Vijay V. Vazirani (Springer)

Reference Books
CS42236:: DISTRIBUTED COMPUTING

Credits: 01  
Teaching Scheme: -Tutorial 1 Hr/Week

Prerequisites: Operating Systems.

List of Contents

A TERM-WORK containing the record of the following:

1. Design and develop a basic prototype distributed system (e.g. a DFS).
2. Design and implement client server application using RPC/ RMI mechanism (Java)
3. Design and implement a clock synchronization algorithm for prototype DS.
4. Implement Ring or Bully election algorithm for prototype DS.
5. Ricart Agrawala’s distributed algorithm for mutual exclusion.
7. Simulate Wait for Graph based Centralized or Hierarchical or Distributed algorithm for deadlock detection.
8. Implementation of 2PC / Byzantine Generals Problem
9. Simulate any one of the Consistency models.

Text Books

4. “Distributed Systems Principles and Paradigms”, Andrew S. Tanenbaum & Maarten Van Steen,
Reference Books

Additional Reading
CS42119:: INFORMATION RETRIEVAL

Credits: 01
Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites: Data Structures, Database Management Systems

List of Contents

1. Boolean model
2. Vector Space model
3. Stemming Algorithms and Thesaurus Construction
4. Index creation: Inverted Files, Signature Files, Suffix Trees and Suffix Arrays
5. Sequential Searching and Pattern Matching techniques
6. Latent Semantic Indexing for Text Classification
7. Design and build an Information Retrieval system using suitable library such as Apache Lucene

Text Books


Reference Books

1. “Information Retrieval”, C. J. Van Rijsbergen, Information Retrieval Group, University of Glasgow, online at http://www.dcs.gla.ac.uk/Keith/Preface.html
CS42113:: DIGITAL SIGNAL PROCESSING

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites::

List of Contents

A TERM-WORK containing the record of the following:

Assignments :

1. Verification of sampling Theorem:
2. Linear Convolution
3. GDE
4. Correlation.
5. Pole Zero Implementation
6. Magnitude and Phase response of a system
7. DFT
8. Circular convolution
9. FFT Algorithms (DIT/DIF)
10. Design IIR filter using analog filter approximations
11. Designing FIR filters using windowing techniques

Students should implement the above assignments in Matlab.

Text Books

Reference Books

CS40113:: BUSINESS INTELLIGENCE AND ANALYTICS

Credits: 01
Teaching Scheme: -Tutorial 1Hr/Week

Prerequisites: Database Management Systems.

List of Contents

A TERM-WORK containing the record of the following:

1. Getting started: understand what business do with their data with a scenario based application using QlikView tool
2. Designing an end to end warehousing solution for a real world scenario involving multi-dimensional modeling, designing data cube, doing ETL, OLAP and reporting
3. Getting started with R
4. Using R for data preprocessing, exploratory analysis, visualization, correlation and regression analysis, hypothesis testing, chi square test
5. Data analysis case study using R for a readily available data set
6. BigData Analytics - MapReduce and exposure to Hadoop, Using R over Hadoop
7. [Optional] A group mini-project: take a real world data analysis problem and solve it using the above learned concepts
   a. Getting Data from varied sources
   b. Data massaging to prepare it for analysis
   c. Generating visualizations to interpret descriptive analysis
   d. Implementing sampling and estimation techniques
   e. Regression analysis on data
   f. Hypothesis testing

Text Books
1. “Data mining and business analytics with R”, Johannes Ledolter, Wiley
2. “Business Analytics” by James R Evans, Pearson

Page 166 of 251

Reference Books

1. “Business Intelligence for Dummies”
5. “Information Dashboard Design: The effective visual communication of data”, Stephen Few, O’Reilly
CS42305:: MOBILE COMPUTING

Credits: 01  Teaching Scheme: - Laboratory  2 Hrs/Week

Prerequisites: Computer Networks

List of Practical

Assignments should be implemented on android operating systems.

1. Design simple GUI application with activity and intents e.g. calculator.
2. Design an application to handle address book and call log.
3. Design an application to display map with current location.
4. Design an application to send SMS and emails.
5. Design a client server application to communicate with database server.

Course Project: The project needs to be done in the group of 2 or 3 where students need to select real world problem and present a solution. During the evaluation of the project the emphasis is on: (a) how problem selection is done and (b) is the application user friendly.

Text Books

Reference Books
CS42327 :: CLOUD COMPUTING

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Computer Networks, web Technology, Operating Systems.

List of Contents

A TERM-WORK containing the record of the following:

1. Hands on virtualization using XenServer
2. Hands on containerisation using Docker
3. Deployment and Configuration options in Amazon (AWS)
4. Deployment and Configuration options in Google Cloud
5. Deployment and Configuration options in Microsoft Azure
6. Building a 'HelloWorld' app for the cloud
7. Deploying the 'HelloWorld' app for the cloud

Text Books
7. Cloud Computing for Dummies by Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper (Wiley India Edition)
9. Cloud Security by Ronald Krutz and Russell Dean Vines, Wiley-India

Reference Books
1. Google Apps by Scott Granneman, Pearson
2. Cloud Security & Privacy by Tim Malhar, S.Kumaraswammy, S.Latif (SPD,O’REILLY)
4. Cloud Computing Bible by Barrie Sosinsky, Wiley India
CS42303:: PARALLEL COMPUTING ON GPU

Credits: 01  
Teaching Scheme: Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

1. Parallel GPU implementation of vector-vector operations
2. Parallel GPU implementation of vector-Matrix operations
3. Parallel computation of binomial coefficient matrix
4. Parallel GPU implementation of Matrix-Matrix operations
5. Assignment focusing on optimization of data transfer between CPU and GPU: using page locked host memory and to avoid the data transfer
6. Assignment focusing on memory optimization: use of GPU shared, constant and texture memory.
7. Parallel GPU implementation involving kernel looping.
8. Parallel computation of set of multi-indices on GPU.

Text Books

2. CUDA by Example: An Introduction to General-Purpose GPU Programming by Jason Sanders and Edward Kandrot
3. Parallel Programming in C with MPI and OpenMP by Michael J. Quinn, Tata McGraw-Hill Edition

References:

http://developer.nvidia.com/
www tutorials on introduction to parallel computing
Other references suggested by instructor
CS42301:: ADVANCED COMPUTER GRAPHICS

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practical

1. Implement an OpenGL program to draw different 2D shapes.

2. Implement an OpenGL program to draw 2 overlapped shapes and use alpha blending.

3. Implement an OpenGL program to draw 3D cube and apply transformations.

4. Implement an OpenGL program to draw 12 spheres and apply different light effects.

5. Implement an OpenGL program to draw scene and apply fog effect.

6. Implement an OpenGL program to draw 3D cube and apply different textures on different faces.

7. Implement CUDA program for the prefix addition.

8. Implement CUDA program for the multiply two matrices.


10. Implement a program for edge detection using Gaussian filter.

Text Books


Reference Books


Additional Reading
Page 171 of 251
CS42331: Enterprise Systems

Credits: 01  
Teaching Scheme: - Laboratory  2 Hrs/Week

Prerequisites: Software Engineering, Database Management System

List of Practical

2. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system along with identification of Business Processes and develop full detail Business Process diagrams.
3. To develop Domain-driven vocabulary of the target system indicating domain lexicon and context-based terminologies.
4. To identify and categorize the target system services with detailed service specifications modeled with component diagram incorporating appropriate architectural style and coupling.
5. To design the service layers and tiers modeled with deployment diagram accommodating abstraction, autonomy, statelessness and reuse.
6. To map the service levels and primitives to appropriate Strategies for data processing using XML / XQuery/ JSON / JAXB.
7. To produce, invoke, compose Web Services using SOAP, WSDL and UDDI.
8. To implement and integrate the components of the target system using .NET / J2EE platforms adhering to Service specifications.
9. To create the balanced scorecard for the target system indicating the standards and principles applied.

Text Books:


Reference Books :

5. Mike Rosen, Boris Lublinsky, Kevin T. Smith, Marc J. Balcer, Applied SOA: Service-Oriented
CS42335: Digital Image Processing

Teaching Scheme: - Lab 2 Hrs/Week

Credits: 03

Prerequisites:

List of Contents
1. RGB to grey conversion
2. Histogram Equalization
3. Smoothing Filters and Sharpening Filters
4. Binarization using adaptive thresholding
5. Perspective Transform
6. Opening and Closing Operations
7. Traffic Sign Recognition

Text Books

Reference Books
CS47303:: PROJECT STAGE 2

Credits: 04

Teaching Scheme: - Practical 8 Hrs/Week

Aim
This course addresses the issues associated with the successful management of a software development project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and their career. The Project Work can lead to:

f. Transform existing systems into conceptual models.
g. Transform conceptual models into determinable models.
h. Use determinable models to obtain system specifications.
i. Select optimum specifications and create physical models.
j. Apply the results from physical models to create real target systems.

Overview of the Course:

9. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
10. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student’s own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.
11. The project must be open-ended – meaning that there is not a known correct answer to the design problem. Students are expected to apply their creativity (simply copying or re-creating something that already exists is not acceptable).
12. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
13. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting Requirement Definition Document, Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.
14. The Project Work will be assessed jointly by a panel of examiners having more than Five Years experience. The Project Groups will deliver the presentation of the Project Work which will be assessed by the panel.

15. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate’s performance based on presentation skills, questions based on the Project Work, understanding of the Project, analysis and design performed for the project.

16. The Student Project Groups are expected to work on the recommendations given by the panel of examiners. In no case any variation in Project Theme will be permitted.

Assessment Scheme

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Content</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Requirement Specification</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>Feasibility Study</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>System Analysis</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>System Design</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Presentation of the Project Work</td>
<td>10</td>
</tr>
</tbody>
</table>

Note:
The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members. The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted. Following is the list of recommended domains for Project Work:

Course Outcomes:
Upon completion of the course, graduates will be able to -

1. Model the Real World Problem
2. Identify the Design within Specification and Available Resources
3. Realise the Solution within Defined references
4. Defend his Design with Technical and Ethical reasoning
5. Adapt to changing Technological and Human resource advances
6. Use the gained knowledge for other Real World Problems
MODULE VIII
<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme ( Hrs. / week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lectures</td>
<td>Tutorial</td>
<td>HA</td>
</tr>
<tr>
<td>CS40112</td>
<td>Software Testing and Quality Assurance</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS40114</td>
<td>Information Systems Security</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS42128</td>
<td>Machine Learning</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS42104</td>
<td>Neural Network</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS42124</td>
<td>Algorithmic Number Theory and Algebra</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS42118</td>
<td>Geographical Information Systems</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS42129</td>
<td>Management Information Systems</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS42121</td>
<td>Advanced Computer Architecture</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>CS42116</td>
<td>Convergence Technologies</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>CS42130</td>
<td>Embedded Systems</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>14</strong></td>
<td><strong>2</strong></td>
<td><strong>5</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Subject Name</th>
<th>Teaching Scheme ( Hrs. / week)</th>
<th>Examination Scheme</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Practicals</td>
<td></td>
<td>ISE</td>
</tr>
<tr>
<td>CS40312</td>
<td>Software Testing and Quality Assurance</td>
<td>2</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Credits</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>CS40314</td>
<td>Information Systems Security</td>
<td>2</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>CS47308</td>
<td>PROJECT STAGE 3 (B.Tech Semester II) Irrespective of Module</td>
<td>12</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>16</td>
<td>70</td>
<td>30</td>
</tr>
</tbody>
</table>

**Abbreviations Used**

- HA: Home Assignment
- TA: Teacher Assessment
- ISE: In Sem. Examination
- ESE: End Sem. Examination
- CA: Continuous Assessment
- #: Alternate Week Laboratory
- ^: Detail Enclosed Separately

**TOTAL CREDITS**: 22
CS40112: Software Testing and Quality Assurance

Credits: 03
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Software Engineering, Database Management System

Unit I: Software Measurement


Part B: The Rayleigh Model, Exponential Distribution and Reliability Growth Models, SRE process, Reliability Concepts: Hardware and Software, Deploying SRE

Unit 2: Principles of Testing


Part B: Analysis of Flow Graphs, Complexity Measures and computations

Unit 3: Functional Testing


Unit 4: Higher Order Testing


Unit 5: Software Quality Assurance (8 Hrs)


Part B: Software Benchmarks and Baselines, Key Factors for Software Assessment and Benchmark Studies, Identifying Software Best and Worst Practices

Text Books:

Reference Books:

Course Outcomes:
Upon completion of the course, graduates will be able to –

1. Judge application of software measurement theory and comprehend evaluative options.
2. Infer problem levels so as to prioritize them based on commonly acceptable evaluation practices.
3. Devise a framework to hypothesize the desirable outcomes by application of suitable testing techniques and strategies.
4. Conform to well documented specification in order to defend the practicability of solution sustainment.
5. Demonstrate the skillset as a tester to neutralize the consequences of wicked problem.
6. Creatively adapt to new segments uprising in testing arena.
CS40108 :: INFORMATION SYSTEMS SECURITY

Credits: 03
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Networks.

Unit 1: Introduction


PART B: SYN-Flooding, SQL-injection, DNS poisoning, Sniffing, Sabotage flashing attack, Security of Embedded Devices.

Unit 2: Private key cryptography

PART A: Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field). Role of random numbers in security, Importance of prime numbers. Data Encryption Standard: Block cipher, Stream cipher, Feistel structure, round function, block cipher modes of operation, S-DES, Attacks on DES, S-AES, AES.

PART B: Chinese remainder theorem, Elementary Ciphers (Substitution, Transposition and their Properties), Frequency analysis.

Unit 3: Public key cryptography


Unit 4: Information Security


PART B: SOA and Cloud data management.

Unit 5: Security application and design


PART B: Cyber laws, Cyber Security, IDS, SNORT, Firewall, Cloud Security Tradeoffs.

Text Books


Reference Books


Additional Reading


Course Outcomes

Upon completion of the course, the students will be able to:

1. Analyze cryptographic techniques using a mathematical approach by examining nature of attack.

2. Establish type of attack on a given system.

3. Simulate different types of attacks using tools.

4. Justify various methods of access control for application of technologies to various sections of industry and society.
5. Design a secure system for protection from the various attacks for 7 layer model by determining the need of security from various departments of an organization.

6. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
CS42120 :: DATA MINING

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Database Management Systems.

Unit 1: Data Mining and Preprocessing (9+2 Hrs)
PART A: Introduction: Need of Data Mining, Knowledge Discovery in Database (KDD), Architecture of Data Mining System; Data Objects and Attribute Types, Statistical Description of Data, Data Visualization
PART B: Data Preprocessing: Central Tendency, Data Dispersion, Data Summarization, Cleaning, Integration, Transformation, Reduction, Discretization, Binarization, Variable Transformation

Unit 2: Mining Frequent Patterns, Association and Correlations (7+1 Hrs)
PART A: Frequent Itemset Mining: Interesting Item Set Mining: Market Basket Analysis, Generating Association Rules, Apriori Algorithm, A pattern growth approach for mining frequent item set, Mining frequent item-sets using vertical data, Evaluation of Association Patterns, From Association Analysis to Correlation Analysis
PART B: Sequential Pattern Mining Algorithms, Pattern mining in multi-level, multi-dimensional space

Unit 3: Classification and Prediction (9+2 Hrs)
PART A: Classification: Decision Tree Classifier, Rule Based Classification, Bayesian Classification, Neural Network Classification: Back Propagation Algorithm, Lazy Learner: kNN Classifier, Support Vector Machine Model Overfitting, Classifier Accuracy Measures, Techniques for Evaluating Classifier Accuracy, Ensemble Methods, Multiclass Problem
PART B: Case-Based Reasoning, Associative Classification, Other Classification Techniques: Genetic Algorithm, Fuzzy Set Approach, Rough Set, Constraints Based Association Mining

Unit 4: Clustering and Outlier Detection (8+2 Hrs)
PART A: Cluster Analysis: Categories of Clustering methods, Different Types of Clusters, Partitioning methods: k-Means, k-Medoids; Hierarchical Clustering Methods: BIRCH, Chameleon; Grid Based Methods: STING; Density based Clustering: DBScan Cluster Evaluation
Outlier Analysis: Types of outlier, Proximity based approach: distance based, Density based approach

PART B: Grid Based Methods: CLIQUE, Density based Clustering: OPTICS, Deviation based outlier detection approach: grid based

Unit 5: Data Mining on different Databases (7+1 Hrs)
PART A: Multimedia Data Mining, Web Mining, Text Mining, Spatial Data Mining, Mining on Social Networks, Multirelational Data Mining.
PART B: Graph Mining, Mining Time – Series Data, Data Mining Applications

Text Books

2. “Introduction to Data Mining”, Pang-Ning Tan, Vipin Kumar, Michael Steinbach, Pearson

Reference Books

CS42128: MACHINE LEARNING

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Artificial Intelligence.

Unit 1: Fundamentals (7 Hrs)
Part-A: Soft computing and hard computing, types of learning, activation functions, Learning Rules, Mc-Culloch-pitts Neuron model, single layer and multilayer perceptron, Backpropagation algorithm, Well posed learning problem, Designing a learning system, Perspectives and issues in soft computing,
Part B: Solving Problems based on above algorithms.

Unit 2: Concept and decision tree Learning (7 Hrs)
Part B: Solving Problems based on above algorithms.

Unit 3: Evaluating Hypothesis (6 Hrs)
Part B: Solving Problems based on above algorithms.

Unit 4: Machine Learning Algorithm
Hidden markov model, Genetic algorithm, SVM, Kernel functions, Linear SVM, Nonlinear SVM, Regression analysis, ensemble classifiers, Validation, evaluation.
Part B: Solving Problems based on above algorithms.

Unit 5: Clustering Algorithm and recurrent Networks (7 Hrs)
Part-A: k-means algorithm, k-nearest neighbor learning, weighted majority algorithm, Hopefield Net, Hamming net, Maxnet, Kohonen self organizing map, Principal component Analysis (PCA), Applications of machine learning.
Part B: Solving Problems based on above algorithms, locally weighted regression,

Text Books
Reference Books
3. Research papers suggested by the faculty.

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Discover computing problem situations for a well-defined learning problem for a given task with different mathematical models used in machine learning.
2. Understand a number of models for supervised, unsupervised, and reinforcement machine learning systematically.
3. Analyze different algorithms according to the properties of their inputs and outputs using different types of big data.
4. Use R programming tool, Matlab and WEKA toolkit for implementation.
5. Minimize errors in the learning and training process.
6. Create new algorithms for the techno crafts and society and evaluate the systems in use using real time big data.
CS42104: NEURAL NETWORKS

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Introduction (8+2 Hrs)
PART A: Introduction and Role of Neural Network (NN), Applications: Signal Processing, Pattern recognition, Medicine, Speech Production/Recognition, Business etc., Artificial Neural Networks, Bilogical Neural Networks, Components of Neural Network, Network topologies, Linear Separability, Hebb Net, Perceptron, Adaline/Madaline, Paradigms of Learning: Unsupervised, Reinforced, Supervised, Competitive, Offline or Online, Activation Functions
PART B: Examples on Learning.

Unit 2: The Perceptron, Backpropagation and other Neural Networks (8+1 Hrs)

Unit 3: Competitive Neural Network (8+2 Hrs)
PART A: Fixed weight Competitive Neural Network, Kohonen Self Organizing Maps, Learning Vector Quantization, Counter propagation: Fully Counter propagation neural network, Forward only Counter propagation Neural network.
PART B: Applications of Self organizing Maps

Unit 4: Adaptive Resonance Theory (ART) (8+1 Hrs)
PART B: Case Study: ART2 Application

Unit 5: Pattern Association (8+2 Hrs)
PART A: Training Algorithm for Pattern association, Heteroassociative Memory Neural Network, Auto associative Neural Network, Iterative Auto associative Neural Network, Discrete Hopfield Network, Bidirectional associative Memory (BAM),
PART B: Perceptron Case Study
Text Books

Reference Books

Additional Reading

Course Outcomes:
Upon completion of the course, graduates will be able to –
1. To Understand a number of models for supervised, unsupervised, and reinforcement neural networks systematically.
2. Student should be able to analyze different algorithms according to the properties of their inputs and outputs using different types of big data.
3. Design the most appropriate neural network for classification, Clustering, automatic detection and optimization.
4. Implement the algorithms in a software environment using MATLAB / Neural ware Professional and R-Programming Language.
5. To evaluate the neural network algorithms
CS42124:: ALGORITHMIC NUMBER THEORY AND ALGEBRA

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Basic introduction to linear algebra and abstract algebra (though it is not presumed) is helpful to better appreciate the contents of the course. First course on algorithms.

Unit 1 : Basic Number Theory  
PART A: Infinitely many primes in Z(some proofs), greatest common divisor, unique factorization of integers, modular arithmetic, (some results on modular arithmetic: Fermat’s little theorem, Wilson’s theorem,... etc), linear congruences and Chinese Remaindering Theorem, linear Diophantine equations, some arithmetic multiplicative functions (e.g. Euler's phi function, Mobius function, divisor function...), Dirichlet convolution, Mobius inversion. Quadratic residues, Legendre symbol, Gauss’ law of Quadratic reciprocity. Distribution of primes, Prime number theorem.
PART B: More Diophantine equations (e.g. Pell’s equation), more results on distribution of primes( proving lower and upper bound(asymptotic) on pi(x), where pi(x) is number of primes upto x)

Unit 2: Basic abstract Algebra  
PART A: Groups, subgroups, Lagrange’s theorem, group homomorphism, quotient groups, normal subgroups, cyclic groups, Abelian groups and their decomposition, rings, ideals, some examples of rings (e.g. polynomial ring F[x], ring of integers modulo n), units and zero devisors in the ring, structure of unit group of Z_n, integral domains, fields, vector spaces. Unique factorization in ring k[x] where k is a field, Principal Ideal Domains(PID).
Basics of finite fields: existence of finite fields of size $p^n$ for $n>0$, $p$ prime, isomorphism of finite fields of same size, structure of multiplicative group, field automorphisms, Frobenius maps.
PART B: Ring of Gaussian integers Z[i] and Lagrange’s four square theorem, analogies between ring of integers and univariate polynomial ring, more properties of finite fields.

Unit 3: Number theoretic algorithms  
PART A: Euclid’s algorithm for GCD, Extended Euclid’s algorithm, fast integer arithmetic, algorithms for modular arithmetic, fast exponentiation, finding modular inverse, finding order of element in a group, finding quadratic non-residues, algorithm for modular square-root, algorithms for discrete-log problem.
Univariate polynomial arithmetic (multiplication, gcd, etc), irreducibility testing for polynomials in F[x] where F is finite field, ring F[x]/(f), modular inverse computation.
PART B: Chinese Remaindering and computing determinant of integer matrix, Euclid’s algorithm and Fibonacci numbers (a detailed analysis of Euclid’s algorithm), Some more

Page 194 of 251
algorithmic questions over finite fields.

Unit 4: Polynomial Factorization (9+2 Hrs)
PART A: Univariate polynomial factorization over finite fields: revisit unique factorization, Randomized polynomial time algorithms:

a. Berlekamp’s algorithm: linear algebra preliminaries for Berlekamp’s algorithm, Frobenius map, the algorithm, resultants and some implications of Berlekamp’s algorithm, parallel algorithm for computing gcd of polynomials.

b. Cantor-Zassenhaus algorithm: distinct degree and equal degree factorization, CRT and Cantor-Zassenhaus algorithm.

Geometry of Numbers and polynomial factorization over rationals: Basics on integer lattices, Minkowski’s theorems, algorithmic questions over integer lattices (Closest Vector Problem and Shortest Vector Problem) Gauss’ algorithm for SVP in 2 dimensions. Lenstra-Lenstra-Lovasz(LLL) approximation algorithm for SVP. Deterministic polynomial time algorithm for factorization of univariate polynomials over rational(using LLL and Hensel lifting lemma)

PART B: Reduction of polynomial factorization problem to root finding using Berlekamp’s algorithm. Recent deterministic sub-exponential time algorithm for the

Unit 5: Primality (7+1 Hrs)
PART A: Pratt’s result: PRIMES in NP intersection coNP, randomized polynomial time algorithms:

a. Miller-Rabin primality test: Carmichael numbers, derandomizing Miller-Rabin test using Riemann Hypothesis, computing a factor of a number given phi(n) where phi is Euler’s totient function.

b. Solovay-Strassen primality test: Jacobi symbol, quadratic reciprocity, Solovay-Strassen primality test.

Prime number theorem and generating large primes (application to RSA crypto system)
PART B: Agarwal-Kayal-Saxena(deterministic polynomial time) primality test.

Text Books
1. Modern Computer Algebra by Joachim von zur Gathen, Jürgen Gerhard (Cambridge)
2. A computational introduction to Number Theory and Algebra by Victor Shoup(Cambridge)
3. A classical introduction to modern number theory by Ireland and Rosen (Springer)
Reference Books

1. Topis in Algebra by I. N. Herstein (Wiley Publishing company)
2. Algebra by Michael Artin (Pearson Prentice Hall)

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. To solve mathematical problems based on elementary number theory
2. To describe various algebraic structures such as groups, rings, fields, vector-spaces and their interrelation
3. To design efficient algorithms for various number theoretic and algebraic computational problems
4. To evaluate various known algorithms for multivariate polynomial factorization over finite fields and field of rationals
5. To describe efficient algorithmic solutions for shortest vector problem over integer lattices and their applications
6. To explain various issues arising in designing efficient algorithms for testing primality of a natural number
CS42118:: GEOGRAPHICAL INFORMATION SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: GIS and Maps (8+1 Hrs)
PART B: Selection of a GIS Application in Various Domains such as Weather Forecasting, Urban Planning, Agriculture, Defense, Network Applications.

Unit 2: Remote Sensing Fundamentals (8+2 Hrs)
PART B: Study of Satellites such as IRS, OCEANSAT-1, IKONOS etc.

Unit 3: Image Processing (8+2 Hrs)
PART B: Study of GIS Hardware and Software required specially for Image Processing.

Unit 4: Spatial Data Modeling and Management (8+2 Hrs)
PART B: Design a Spatial Database for a Selected Application.

Unit 5: Data Input, Quality and Analysis (8+1 Hrs)

PART B: Identification of Data Inputs Outputs and Study of Required Analytical

Text Books

Reference Books

Additional Reading

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Differentiate among map projections for geographical areas
2. Design spatial data model for geographical area encompassing related entities
3. Create database for geographical area encompassing related entities using GIS software
4. Propose effective remote sensing based solutions addressing requirements in domains such as urban planning, weather forecasting, defense and land management
5. Display the impact of data processes such as data input, quality, analysis and output in GIS applications
6. Build skill set required for responsible positions such as GIS Analyst, GIS Technician and GIS Specialist
CS42129: MANAGEMENT INFORMATION SYSTEMS

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites:

Unit 1: Foundations of Information Systems (8+2 Hrs)

Unit 2: Manufacturing and Service Systems (8+2 Hrs)

Unit 3: e-Business (8+2 Hrs)

Unit 4: Information Systems for Decision Support (8+2 Hrs)
Unit 5: Challenges Ahead (8+2 Hrs)


PART B: Study of Cybercrimes and Preventive Measures w. r. t. Selected Domain.

Text Books


Reference Books


Additional Reading


Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Estimate the functional complexities in manufacturing and service sectors for implementation of Management Information Systems
2. Differentiate management information systems based on their features and applicability
3. Initiate ethically responsible behavior as a professional
4. Respond positively to cultural, political and economical organizational challenges
5. Build a set of skills required for responsible positions such as System Analyst, Business Consultant and Information System manager
6. Follow required domain-specific processes and standards for management information systems

Page 200 of 251
CS42121: ADVANCED COMPUTER ARCHITECTURE

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Organization

Unit 1: Overview of Parallel Processing (8+1 Hrs)

PART A: Overview of Parallel Processing and Pipelining Processing, study and comparison of uni-processors and parallel processors.
Necessity of high performance, Constraints of conventional architecture, Parallelism in uni-processor system, Evolution of parallel processors, future trends, Architectural Classification, Applications of parallel processing, Instruction level Parallelism and Thread Level Parallelism

PART B: Explicitly Parallel Instruction Computing (EPIC) Architecture, Performance Metrics and Measures, Speedup Performance Laws.

Unit 2: Pipelining Processing (8+1 Hrs)

PART A: Principles and implementation of Pipelining, Classification of pipelining processors, Pipeline Architecture, Study and comparison of processors with and without pipelining. General pipelining reservation table, Design aspect of Arithmetic and Instruction pipelining, Pipelining hazards and resolving techniques, Data buffering techniques, Job sequencing and Collision, Advanced pipelining techniques, loop unrolling techniques, out of order execution, software scheduling, trace scheduling.

PART B: Advances in pipeline architectures. Implementation issues of a program on any pipelined processor their analysis.

Unit 3: SIMD Computer Organization and Parallel Algorithms For Array Processors (8+1 Hrs)


PART B: Implementation issues of Matrix multiplication and sorting on array processor and their analysis.

Unit 4: Multiprocessor (8+1 Hrs)
PART A: Microprocessor Architectures, study and comparison of Loosely and Tightly coupled multiprocessors.
Loosely and Tightly coupled multiprocessors, Processor characteristics of multiprocessors, Inter Processor communication network, Time shared bus, Crossbar switch, Multiport Memory Model, Memory contention and arbitration techniques, Cache coherency and bus snooping, Massively Parallel Processors (MPP), Cow’s and NOW’s Cluster and Network of Work Stations), Chip Multiprocessing (CMP).

PART B: Implementation issues of a program on multiprocessor system

Unit 5: Parallel Programming Techniques (8+1 Hrs)
Message passing program development, Synchronous and asynchronous message passing, Message passing parallel programming, Shared Memory Programming, Data Parallel Programming.

PART B: Implementation issues of a multithreaded program.

Text Books

Reference Books
1. Rajaraman, L Sivaram Murthy, "Parallel Computers", PHI.
5. Richard Y. Kain, Advanced Computer Architecture

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Describe the principles of computer design.
2. Improve the performance of applications of on modern and high performance computers
3. Compare the performance of different architectures.
4. Develop application for high performance computing systems.
5. Design solutions to computing problems using alternative architectures.
6. Analyze architectures performance and select among different ones for particular use scenarios.
CS42116 :: CONVERGENCE TECHNOLOGY

Credits: 03  
Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Computer Networks.

Unit 1: Introduction to Convergence (7+2 Hrs)
PART A: what is network Convergence, the promise of network convergence, networking issues and convergence, Voice and data network characteristics, benefits of IP centric network, challenges of converged network, introduction to VOIP, applications of converged networks, VOIP implementation challenges.

PART B: voice and data network growth factor, effects of network convergence on businesses.

Unit 2: Protocols and Standards for Convergence (9+1 Hrs)

PART B: MGCP, Audio and Video Codecs.

Unit 3: Switching networks (9+1 Hrs)
PART A: ISDN: conceptual view of ISDN, transmission structure, user-network interface configuration, ISDN Protocol Architecture, ISDN connection, Addressing, Interworking, PRI, BRI, LAPD, Basic Call control, SS7.

B-ISDN standards, Broadband services, B-ISDN architecture, B-ISDN protocol reference model.

PART B: ISDN standards, SDH.

Unit 4: Frame Relay and SMDS (7+1 Hrs)
PART A: Frame Relay Circuits, Frame mode protocol architecture, frame mode call control, LAPF, Congestion in frame relay networks, approaches to congestion control, Traffic rate management, Explicit congestion avoidance, implicit congestion control.

SMDS: introduction to SMDS, SMDS interface protocol, SMDS addressing.

PART B: Comparison of SMDS with other LAN technologies.

Unit 5: ATM technology (8+1 Hrs)
PART A:  ATM VPI& VCI, Creation of virtual channel, Definitions of Virtual circuit and permanent virtual circuit, ATM reference model, step-by-step PVC example of how ATM network processes cells, AAL, Adaption layer from voice over ATM perspective AAL1,AAL2, AAL3, Connection admission control (CAC). Cell Loss Priority (CLP), ATM and convergence technology.

PART B:  ATM versus Frame relay, ATM versus SONET.

Text Books

Reference Books

Course Outcomes:
Upon completion of the course, the graduate will be able to:
1. Categorize voice and data networks based on various protocols.
2. Analyze the protocols and standards for converged networks.
4. Design the converged network to fulfill the societal requirement.
5. Judge the impact and benefits of converged network in exploitation on environment and society.
6. Prepare cost effective solutions to fulfill the need of convergence technology.
CS42130 :: EMBEDDED SYSTEMS

Credits: 03  Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Understanding of Microprocessors, Peripheral Chips, Conversion, Interfacing Techniques.

Unit I  Introduction to Embedded Systems (8+1 Hrs)
PART B: Memory Selection Criterion

Unit 2: Bus Design (8+1 Hrs)
PART B: Study of RS 232C, Centronics and USB

Unit 3: Microcontroller Support Units (8+1 Hrs)
PART A: Details of Components of Embedded Systems - Management of Power Supply, Clocking Unit, Real Time Clock and Timers, Reset Circuitry and Watchdog Timer. Structural Units of Processor, Processor and Memory Selection, Memory Map Of Embedded System, Interfacing Processors, Memories and I/O. Processor, Memory Map Of Embedded System
PART B: Design Parameters problem solving. PWM ADCTimers

Unit 4: I/O Interfacing (8+1 Hrs)
A I/O interfacing and Communication I/O devices, ADC / DAC, Optical Devices such as LED / LCD Display devices, Opto-Isolator, Relay & stepper motor, Timers/Counters. Parallel v/s serial communication. Parallel ports their uses in device interfacing.
PART B: Design Implementation of OP Amps

Unit 5: RTOS (8+1 Hrs)
PART B: Case Study of Embedded Systems in Detail. (H/W + S/W Algo)

Text Books

Reference Books

Additional Reading
1. Microcontroller Handbook
6. Tammy Noergaard, — Embedded Systems Architecture by Elsevier

Course Outcomes:
Upon completion of the course, graduates will be able to –

2. Design system interconnects for effective throughput.
3. Create designs using Simulation and RTOS Tools.
5. Cooperate with diverse Teams for delivering automation Solutions.
6. Adapt to ever changing technological Advances.
CS42120 :: DATA MINING

Credits: 01  
Teaching Scheme: -Tutorial: 1 Hr/Week

Prerequisites: Database Management Systems.

List of Contents

A TERM-WORK containing the record of the following assignments:

Implementation of following data-mining tasks using a suitable Data Mining Toolkit:
1. Data Preprocessing
2. Association Mining Algorithms
3. Classification Algorithms
4. Linear Regression
5. Clustering Algorithms

Text Books
2. “Introduction to Data Mining”, Pang-Ning Tan, Vipin Kumar, Michael Steinbach, Pearson

Reference Books
CS42128 :: MACHINE LEARNING

Credits: 01  
Teaching Scheme: -Tutorial: 1 Hr/Week

List of Tutorials

1. Implementation of learning algorithms like Find S algorithm, Version space and the candidate elimination algorithm, list then eliminate algorithm for simple real world problems.
2. Implementation of learning algorithms like Backpropagation algorithm, Support Vector Machines for real time problems.
3. Implementation of algorithms like Evaluating hypothesis accuracy, Sampling theory, Central limit theorem, hypothesis testing, for real time problems.
4. Implementation of learning algorithms like Bayesian Learning for real time problems.
5. Implementation of learning algorithms like weighted majority algorithm, Instance-Based Learning: k-nearest neighbor learning, locally weighted regression for real time problems.
7. Implementation of learning algorithms like unsupervised or reinforcement learning for real time problems.
8. Implementation of Fuzzy logic for real time problems.
9. Presentations on selected research topics.
10. A mini project for a real problem.

Text Books

2) Randy L. Haupt and Sue Ellen Haupt, Practical Genetic Algorithms.

Reference Books

3) Research papers suggested by the faculty.

Outcomes:

At the end of the course, the students will be able to:
1. Use the advanced concepts and techniques of soft computing.
2. Develop skills of using recent soft computing software for solving practical problems.
3. Gain experience of doing independent study and research.
4. Able to Publish papers in international conferences and journals.
CS42104 :: NEURAL NETWORKS

Credits: 01

Teaching Scheme: -Tutorial: 1 Hr/Week

List of Tutorials:

Minimum eight experiments based on the following topics.
1. Supervised Learning rules for a single neuron
2. Unsupervised Learning rules.
3. Simple Perceptron classifier
5. Backpropagation algorithm
6. MAXNET
7. Hamming Distance Classifier
8. Hopfield network
9. SOM
10. SVM
11. Verification of logic gates using NN algorithms
12. Feature extraction for a given real world problem of importance.
14. Two short assignments based on research papers suggested by staff.

Text Books

3. Prof. Dr. S. T. Patil, Neural Networks and applications, Nirali Prakashan, Pune.

Reference Books

5. John Yen and Reza Langari, “Fuzzy logic, intelligence, control and information”.
7. Some basic and modern research papers suggested by staff in charge.
CS42124:: ALGORITHMIC NUMBER THEORY AND ALGEBRA

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Basic introduction to linear algebra and abstract algebra (though it is not presumed) is helpful to better appreciate the contents of the course. First course on algorithms.

List of Contents

A TERM-WORK containing the record of the following:

1. Problem solving based on topic like prime numbers, gcd of integers, some simple number theoretic questions
2. Modular arithmetic, Chinese remaindering applications
3. Multiplicative functions, analytical estimates of some number theoretic functions
4. Problem solving (based on topics Groups, subgroups, homomorphism, cyclic groups etc)
5. Ring, ideals, some examples of commutative rings
6. Vector spaces
7. Integral domains, finite fields
8. More on finite fields, automorphisms, Frobenius maps, some algorithmic questions...
9. Ring of univariate polynomials F[x], quotient ring F[x]/(f)
10. Berlekamp’s algorithm and more
11. Integer lattices
12. Carmichael numbers, Lagrange symbol, Jacobi symbols, Primality testing.

Text Books

1. Modern Computer Algebra by Joachim von zur Gathen, Jürgen Gerhard (Cambridge)
2. A computational introduction to Number Theory and Algebra by Victor Shoup (Cambridge)
3. A classical introduction to modern number theory by Ireland and Rosen (Springer)

Reference Books

1. Topics in Algebra by I. N. Herstein (Wiley Publishing company)

2. Algebra by Michael Artin (Pearson Prentice Hall)
CS42118::GEOGRAPHICAL INFORMATION SYSTEMS

Credits: 01

Teaching Scheme: - Tutorial 1 Hrs/Week

Prerequisites:

List of Practical

A TERM-WORK containing the record of the following:

1. Prepare a map for the selected geographical area as per topological survey.
2. Design a spatial database for the entities related with a geographical area specified in above map.
3. Study the connectivity of maps with spatial databases.
4. Analyze a case study of any GIS application of your choice.
5. Prepare a presentation on any latest GIS technology / technique / software /hardware.

Text Books


Reference Books

CS42129: MANAGEMENT INFORMATION SYSTEMS

Credits: 01  
Teaching Scheme: - Tutorial 1 Hr/Week

List of Contents

A TERM-WORK containing the record of the following:

1. Consider any organization from any sector. Study its organizational structure and comment about it.
2. By giving examples, differentiate between operational, strategic and tactical level of management process and its effect on design information system.
3. Identify and evaluate the design considerations for the given information system.
4. Study testing and quality assurance strategies.
5. Identify cases of computer crime, hacking, and cyber theft with respect to given information system. Plan about how to avoid and deal with such kind of security threats.

Text Books


Reference Books


Additional Reading

CS42121 :: ADVANCED COMPUTER ARCHITECTURE

Credits: 01
Prerequisites: Computer Organization
Teaching Scheme: - Tut 1 Hrs/Week

List of Contents:

1. Introduction to Explicitly Parallel Instruction Computing (EPIC) Architecture.
2. To study numerical based on Performance Metrics and Measures, Speedup Performance Laws.
3. To study implementation issues of a program on any pipelined processor and their analysis.
4. To study numerical based on Pipelining.
5. To study implementation issues of Matrix multiplication and sorting on array processor and their analysis.
6. To study implementation issues of a program on multiprocessor system.
7. To study implementation issues of a multithreaded program.
8. To study Architecture of Multithreaded processors

Text Books


Reference Books

1. Rajaraman, L Sivaram Murthy, "Parallel Computers", PHI.
5. Richard Y. Kain, Advanced Computer Architecture
CS42116 :: CONVERGENCE TECHNOLOGY

Credits: 01      Teaching Scheme: -Tutorial: 1 Hr/Week

Prerequisites: Computer Networks.

A TERM-WORK containing the record of the following:

A. Assignments :
   1. Study of audio and video data.
   2. Signal passing through network for voice data.
   4. Study of gatekeepers in convergence technology.
   5. Study of ISDN devices.
   7. Study of SS7 implementation.
   8. Comparison of ISDN and B-ISDN.
   9. Use of frame relay in convergence.
  10. Examples on convergence

Text Books

Reference Books

Additional Reading
CS42130:: EMBEDDED SYSTEMS

Credits: 01
Teaching Scheme: -Tutorial 1 Hr/Week

Prerequisites: MI MPMC.

List of Contents

10. Interface the LCD to Microcontroller
11. Understanding Different Interrupts and Programming The Interrupts
12. Programming the ADC in Microcontroller
13. Program the USART in Microcomputer.
14. Program the I2C Interface
15. Program the SPI interface.
16. Study different kinds of RESET.
17. Program the Timers for Creating the Square Wave.
18. Switching TRIACS and SCR.
19. Mini Project Design and Implementation

Text Books

Reference Books

Additional Reading
Page 219 of 251
6. Tammy Noergaard, — Embedded Systems Architecture by Elsevier
CS40312: Software Testing and Quality Assurance

Credits: 01
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Software Engineering

List of Contents

A TERM-WORK containing the record of the following:

1. To Prepare Test Plan for the implemented system under test. The Test plan consists of following issues.
   a. Purpose of the test.
   b. Location and schedule of the test.
   c. Test descriptions.
   d. Pass and Fail Criteria.

2. To identify and narrate Test cases, Test scripts/procedures and Test incident Report identifier for the system under test. Refer Use case analysis document to prepare mentioned/identified test documents.

3. To perform Unit testing especially indicating the traced Independent data paths, Control paths and Error handling paths. Prepare control flow graphs for the unit under test. Compute the Cyclomatic complexity of the unit.

4. To perform Data Flow testing for the Program Segments by identifying the Definition-Use chain and type of data flow anomaly.

5. To perform Mutation Analysis of the Program Segments along with mutant history, mutation score and type of mutation by using any Code analysis Tool / Mutation Testing Tool (JUNIT, MuJava).


7. To perform Black-Box Testing for all the units contained in the architectural segments using Equivalence Partitioning, Boundary Value Analysis and Orthogonal Array testing methods. To study exploratory Testing for the Module under Test and merits/demerits of this technique.

8. To perform Regression Testing / GUI Testing of the System under construction with Unit and Integration profiles by using any Functional Testing Tool.

Text Books:


Reference Books:

CS40314 :: INFORMATION SYSTEMS SECURITY

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

List of Practicals

1. Demonstrate: SQL injection, Cross-site scripting, buffer overflow.

2. Implement packet sniffer.

3. Implementation of Caesar and Vigenere Cipher

4. Implementation of Playfair Cipher

5. Implementation of Hill Cipher

6. Implementation of S-DES

7. Implementation of S-AES

8. Implementation of RSA.

9. Implementation of Diffie-Hellman key exchange

10. Implementation of ECC algorithm.

Text Books


Reference Books


Additional Reading
CS47308:: PROJECT STAGE 3

Credits: 06

Aim
This course addresses the issues associated with the successful management of a software development project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and their career. The Project Work can lead to:

- a. Transform existing systems into conceptual models.
- b. Transform conceptual models into determinable models.
- c. Use determinable models to obtain system specifications.
- d. Select optimum specifications and create physical models.
- e. Apply the results from physical models to create real target systems.

Overview of the Course:
1. The Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents:

   (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

<table>
<thead>
<tr>
<th>Sr.</th>
<th>Project Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Cover Front Page</td>
</tr>
<tr>
<td>2</td>
<td>Project Completion Certificate [Institute]</td>
</tr>
<tr>
<td>3</td>
<td>Project Completion Letter [In case of Sponsored Projects]</td>
</tr>
<tr>
<td>4</td>
<td>Acknowledgments</td>
</tr>
<tr>
<td>5</td>
<td>Table of Contents</td>
</tr>
<tr>
<td>6</td>
<td>List of Figures</td>
</tr>
<tr>
<td>7</td>
<td>List of Tables</td>
</tr>
<tr>
<td>8</td>
<td>Project Synopsis [Problem Background, Existing System Details, Proposed Solution]</td>
</tr>
</tbody>
</table>
2. The Project Work will be assessed jointly by a panel of examiners consisting of faculty and industry experts. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.

3. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate’s performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

Note:
The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

Assessment Scheme

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Content</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Requirement Specification</td>
<td>05</td>
</tr>
<tr>
<td>2</td>
<td>Feasibility Study</td>
<td>05</td>
</tr>
<tr>
<td>3</td>
<td>System Analysis</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>System Design</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>System Implementation</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>System Testing</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>Presentation of the Project Work</td>
<td>20</td>
</tr>
</tbody>
</table>

Page 226 of 251
Course Outcomes:
Upon completion of the course, graduates will be able to:

1. Identify the Design within Specification and Available Resources
3. Lay Down rules to Minimise Adverse Impact of designed Solutions
4. Foresee the Impact of Design Implementation
6. Adhere to rigorous Standards laid down by Professional Engineering Bodies
CS33303:: ADVANCED JAVA

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Core Java.

List of Practical

1. Design a java application to demonstrate the use Java revision, anonymous inner classes, file handling, GUI, event handling, debugging using IDE.

2. Design a java application to demonstrate use of Multithreading, concurrency, synchronous and asynchronous callbacks, ThreadPools using ExecutorService.

3. Design a java application to demonstrate use of Collections and generics.

4. Design a java database application using multithreading and concurrency control.

5. Design a java application to demonstrate use of Servlets and JSP.

6. Design a client-server application demonstrating the use of Java I/O using sockets with GUI for configurations.

7. Design a java RMI application.

8. Designing a java application to demonstrate use of Web Services - REST and SOAP.

9. Design a java application to demonstrate dynamic invocation using reflection.

Reference Books


Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Select the advanced features of java in solving a complex problem.
2. Implement appropriate exception handling in code.
3. Choose the appropriate advanced java features depending on problem statement.
4. Practice an IDE like Eclipse or Netbeans for quicker coding/debugging.
5. Produce reusable and extensible design to minimise rework.
6. Construct the solution by breaking the complex problem into smaller problems.
CS33312:: PIC Microcontroller

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Microprocessor, x86, x86 Interfacing Chips.

List of Practical

1. Assignment on Program Compilation and Burning into Microcontroller.
2. Assignment on Input Output.
3. Assignment on Interrupt.
4. Assignment on LED.
5. Assignment on Timer.
6. Assignment on LCD.
7. Assignment on UART.
8. Assignment on Write and Read from EEPROM.
9. Assignment on ADC.
10. Assignment on PWM.
11. Assignment on Stepper Motor.

Text Books
1. Data Sheet www.microchip.com
2. Hitachi Data Sheet on LCD HD 44780

Reference Books
1. Microchip 18F45xx

Course Outcomes:

Upon completion of the course, graduates will be able to –

2. Utilize the Structures to effectively solve Computing Problems.
3. Design system interconnects for effective throughput.
4. Validate design outputs using standards test equipment.
5. Design Effective Automation Solutions.
Cooperate with diverse teams for delivering automation Solutions.
CS33313:: MOBILE APPLICATION DEVELOPMENT

Credits: 01  
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Java.

List of Practical

1. Download, Install and Configure Eclipse IDE with Android Development Tools (ADT) plug-ins and Android SDK or Android Studio or Net Beans with Android plugin.

2. Building Simple User Interface using UI Widgets such as Buttons, Text Fields and View.

3. Design an android based application using content provider.

4. Develop an android based application to implement the sequential and random file operation.

5. Develop an android based application to create simple embedded database for the student attendance and find defaulters in the class using SQLite.

6. Design an android based application to demonstrate GPS services using Google map.

7. Design an android based application to implement HTTP operations for internet communication.

8. Design an android based application to implement chat application using socket programming.

9. Design an android based application to take a snapshot by using the Camera in your mobile. Save the snapshot in the image or video format. Use Camera Media API provided Android.

10. Mini Project.

Text Books


Reference Books


Course Outcomes:

Upon completion of the course, graduates will be able to
1. Use embedded database SQLite, Flat files and Multi Media files.
2. Display the current location of a device using google map.
3. Develop the user interface.
4. Choose suitable software tools and APIs for the development of Mobile Application
5. Design and deploy mobile application using software development environment
6. Demonstrate internet based application.
CS3306: ETHICAL HACKING AND NETWORK DEFENSE

Credits: 01  Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Fundamentals of IT, Networking, Microsoft OS, LINUX or UNIX operating systems.

List of Practical
1. Study of different type of attacks
2. Study of Ethical hacking, types of hacking, different phases involved in hacking.
3. Study of skills to become ethical hacker.
4. Study of spoofing techniques
5. Study of password cracking techniques
7. Study of spyware technology
8. Study of types of viruses, antivirus techniques and virus detection mechanism
9. Study of Sniffing techniques and tools.
10. Study of Flooding attacks like MAC flooding, SYN flooding etc.
11. Study of Session Hijacking and prevention of session hijacking.
12. Web based password cracking techniques
14. Study of Physical security.

Text Books
Michael T Simpson – “Ethical Hacking and Network Defense”.

Course Outcomes
Upon completion of the course, the students will be able to:

1. Analyze nature and type of attack.

2. Establish type of attack on a given system.

3. Simulate different types of attacks using tools.
4. Differentiate between the type of communication services used for attack.

5. Design a secure system for protection from the various attacks by determining the need of security from various departments of an organization.

6. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
CS33310: SPRING FRAMEWORK

Credits: 01  
Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Java, JSP, Servlets

List of Practical

1. Assignment on Spring Environment Setup.
2. Assignment on Spring Hello World Example.
3. Assignment on Spring IOC Container.
4. Assignment on Spring Bean Scopes, Spring Bean Life Cycle.
5. Assignment on Spring Bean Post Processors.
6. Assignment on Spring Dependency Injection, Spring Injecting Inner Beans, Spring Injecting Collection, Spring Beans Auto-Wiring.
7. Assignment on Spring Annotation Based Configuration, Spring Java Based Configuration.
8. Assignment on Event Handling in Spring.
9. Assignment on Spring AOP Assignments.
10. Assignment on Spring JDBC assignments.
11. Assignment on Spring Web-MVC Assignments.

Text Books

1. Spring Recipes – A problem solution approach by Gary Mak, Josh Long and Daniel Rubio.
2. Professional Java Development with the Spring Framework, by Rod Johnson

Reference Books

1. Pro Spring 3.0 by Clarence Ho, Rob Harrop.
2. Expert Spring MVC and Web Flow by Seth Ladd, Darren Davison, Steven Devijver, Colin Yates

Course Outcomes:

Upon completion of the course, graduates will be able to -

2. Construct formalized design patterns to effectively implement Java Enterprise Application lifecycle.
3. Create application using Spring Tool Suite, Software project management and comprehension tool like Maven.
4. Demonstrate that the business rules and validations are implemented in shorter time using this framework.
5. Acquire skills to work on real time projects in industry.
6. Use pre-built framework for rapid application development using Spring Framework MVC Applications.
CS33311: STRUTS FRAMEWORK

Credits: 01  
Teaching Scheme: - Laboratory  2 Hrs/Week

Prerequisites: Java, JSP, Servlets

List of Practical

1. Building a Simple Struts Application
2. Struts validator framework
3. Setup validator framework in Struts
4. Struts validator Framework
5. Using the validator framework in struts
6. Validator framework work in Struts
7. Sing validator framework work in struts
8. Using the validator Framework
9. Fixed Value check using struts validator framework
10. Struts 2 double validator
11. Struts 2 Date validator
12. Client Side Address Validation in Struts
13. Struts 2 RequiredString validator
14. Struts 2 E-mail Validator
15. XML files used in Validator Framework?
16. struts - Framework
17. Struts 2 Validation (Int Validator)
18. Struts 2 Url Validator
19. Validation using validator-rules.xml – Struts

Text Books

1. "Jakarta Strus Live" by Rick Hightower published by SourceBeat.

Reference Books


Course Outcomes:

Upon completion of the course, graduates will be able to -

4. Demonstrate that the business rules and validations are implemented in shorter time using this framework.
5. Acquire skills to work on real time projects in industry.
6. Incorporate best practices for building applications with Struts.
CS33314:: PROBLEM SOLVING AND PROGRAMMING

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Data structures.

List of Practical

1. Data structure review (stack, queue, linked list).
2. Graph searching techniques (DFS, BFS, IDDFS etc.) and applications of graph searching in problems in programming competition.
3. Advanced data structures union-find (including optimized algorithms like path compression), segment trees, interval trees, augmented data structures and their applications.
4. String searching algorithms.
5. Dancing links to speed up backtracking.

Text Books


Reference Books


Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Apply and practice logical ability to solve the problems.
2. Modularize the problems into small modules and then convert them into
3. Analyze algorithms and determine their time complexity.
4. Trace and code recursive programs.
5. Choose appropriate problem solving technique
6. Verify and validate the correctness of the algorithm.
CS33315:: BIG DATA TECHNOLOGIES

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites: Programming Skills

List of Practical

1. Study of Hadoop 1 / Hadoop 2 (YARN)
2. Study of hadoop distributed file system (HDFS)
3. Manipulation of data on HDFS
4. Learning Map Reduce Programming
5. Word count problem using Map Reduce Programming
6. Hands-on over Pig
7. Hands-on over Hive
8. Introduction to Hbase

Text Books

2. “Programming Pig”, Allen Gates, O'Reilly

Reference Books

1. “Programming Hive”, Dean Wampler, O'Reilly
2. “HBase: The Definitive Guide”, Lars George, O'Reilly

Additional Reading

1. "Hadoop In Action", Chuck Lam, Manning Publication

Course Outcomes:

Upon completion of the course, graduates will be able to -

1. Illustrate architecture of Hadoop
2. Break down a computing problem into multiple parallel tasks
3. Explain Hadoop Ecosystem
4. Organise input data to handle it using HDFS
5. Apply map reduce programming technique to address real world problems
6. Adapt to upcoming technologies for management of complex big data problems
CS3307:: MATLAB

Credits: 01
Teaching Scheme: - Laboratory 2 Hrs/Week

List of Practical (Any Ten)

1. Introduction to MATLAB, MATLAB Elements & Simple Programs and debugging concepts.

2. Write a Matlab Program for functions.

3. Write a Matlab Programs by using IF Then Else, Case, Statement, for Loop, While loop.

4. Write a Matlab Program for 2-D graph.

5. Write a Matlab Program for 3-D graph.

6. Write a Matlab Program for various Image operations.

7. Write a Matlab Program for Animations.

8. Study of MATLAB debugging commands.

9. Write a Matlab Program to create GUI.

10. Write a Matlab Program to simulate a simple circuit.

11. Write a Matlab Program to create Movie.

12. Write MATLAB Program to read sound file and adjust its parameters.

13. Write MATLAB Program to read .avi file.

Text Books

Reference Books
Course Outcomes:

1. Upon completion of the course, graduates will be able to -
2. Solve Mathematical equations.
3. Design GUI by using MATLAB.
5. Validate design outputs using standards test equipments.
6. Develop animation programs by using MATLAB.
7. Perform various operations on Image.