Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to University of Pune)

Structure & Syllabus of

B.Tech. (Computer Engineering)

Pattern ‘A-14’
Effective from Academic Year 2015-16

Prepared by: - Board of Studies in Computer Engineering
Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Signed by

Chairman – BOS       Chairman – Academic Board
## Content

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Program Educational Objectives and Program Outcome of B.Tech</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>(Computer Engineering)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Course Structure - Module I and II</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>CS10102 Computer programming (Theory Course)</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>CS10302 Computer Programming (Laboratory)</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Course Structure - Module III</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>Course Syllabi for Courses - Module III</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>CS20117 Discrete Structures and Graph theory(Theory Course)</td>
<td>23</td>
</tr>
<tr>
<td>4.2</td>
<td>CS20111 Data Structures (Theory Course)</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>CS20108 Computer Organization (Theory Course)</td>
<td>29</td>
</tr>
<tr>
<td>4.4</td>
<td>CS20116 Problem Solving and Programming (Theory Course)</td>
<td>31</td>
</tr>
<tr>
<td>4.5</td>
<td>CS20113 Digital Electronics and Logic Design (Theory Course)</td>
<td>34</td>
</tr>
<tr>
<td>4.6</td>
<td>CS20216 Problem Solving and Programming (Tutorial)</td>
<td>36</td>
</tr>
<tr>
<td>4.7</td>
<td>CS20113 Digital Electronics and Logic Design (Tutorial)</td>
<td>38</td>
</tr>
<tr>
<td>4.8</td>
<td>CS20311 Data structures (Laboratory Course)</td>
<td>39</td>
</tr>
<tr>
<td>4.9</td>
<td>CS20313 Digital Electronics (Laboratory Course)</td>
<td>40</td>
</tr>
<tr>
<td>4.10</td>
<td>CS27401 Mini Project</td>
<td>42</td>
</tr>
<tr>
<td>4.11</td>
<td>CS24306 PHP MYSQL OR</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>CS24303 C#.NET (Skill Development Course)</td>
<td>44</td>
</tr>
<tr>
<td>4.12</td>
<td>Elective –Soft Skills</td>
<td></td>
</tr>
<tr>
<td>4.13</td>
<td>CS20401 Comprehensive Viva Voce</td>
<td></td>
</tr>
<tr>
<td>4.14</td>
<td>Institute Elective</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Course Structure - Module IV</td>
<td>47</td>
</tr>
<tr>
<td>6</td>
<td>Course Syllabi for Courses - Module IV</td>
<td>49</td>
</tr>
<tr>
<td>6.1</td>
<td>CS21104 Mathematical Transformations and Applications (Theory Course)</td>
<td>49</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Name</td>
<td>Credits</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>6.2</td>
<td>CS20110 Computer Graphics (Theory Course)</td>
<td>52</td>
</tr>
<tr>
<td>6.3</td>
<td>CS20105 Principles of Programming Languages (Theory Course)</td>
<td>54</td>
</tr>
<tr>
<td>6.4</td>
<td>CS21112 Data Communication (Theory Course)</td>
<td>56</td>
</tr>
<tr>
<td>6.5</td>
<td>CS20114 Microprocessor and Interfacing (Theory Course)</td>
<td>58</td>
</tr>
<tr>
<td>6.6</td>
<td>CS20114 Microprocessor and Interfacing (Tutorial)</td>
<td>60</td>
</tr>
<tr>
<td>6.7</td>
<td>CS21104 Mathematical Transforms and Applications (Tutorial)</td>
<td>62</td>
</tr>
<tr>
<td>6.8</td>
<td>CS20314 Microprocessor and Interfacing (Laboratory Course)</td>
<td>64</td>
</tr>
<tr>
<td>6.9</td>
<td>CS20305 Principles of Programming Languages (Laboratory Course)</td>
<td>67</td>
</tr>
<tr>
<td>6.10</td>
<td>CS20310 Computer Graphics (Laboratory Course)</td>
<td>69</td>
</tr>
<tr>
<td>6.11</td>
<td>CS27402 Mini Project</td>
<td>70</td>
</tr>
<tr>
<td>6.12</td>
<td>CS24302 ASP.NET OR</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>CS24304 Python (Skill Development Course)</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>CS24307 Ruby Programming</td>
<td>75</td>
</tr>
<tr>
<td>6.13</td>
<td>Elective – Health &amp; Hobby</td>
<td>-</td>
</tr>
<tr>
<td>6.14</td>
<td>CS20402 Comprehensive Viva Voce</td>
<td>-</td>
</tr>
</tbody>
</table>

**7**

**Course Structure - Module V**

**8**

**Course Syllabi for Courses - Module V**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>CS30101 Operating Systems (Theory Course)</td>
<td>80</td>
</tr>
<tr>
<td>8.2</td>
<td>CS30116 Computer Networks (Theory Course)</td>
<td>83</td>
</tr>
<tr>
<td>8.3</td>
<td>CS30105 Theory of Computation (Theory Course)</td>
<td>86</td>
</tr>
<tr>
<td>8.4</td>
<td>CS31113 Microprocessors and Microcontrollers (Theory Course)</td>
<td>89</td>
</tr>
<tr>
<td>8.5</td>
<td>CS31115 Advanced Data Structures (Theory Course)</td>
<td>91</td>
</tr>
<tr>
<td>8.6</td>
<td>CS30101 Operating Systems (Tutorial)</td>
<td>93</td>
</tr>
<tr>
<td>8.7</td>
<td>CS31215 Advanced Data Structures (Tutorial)</td>
<td>95</td>
</tr>
<tr>
<td>8.8</td>
<td>CS30303 Operating Systems (Laboratory Course)</td>
<td>96</td>
</tr>
<tr>
<td>8.9</td>
<td>CS30316 Computer Networks (Laboratory Course)</td>
<td>98</td>
</tr>
<tr>
<td>8.10</td>
<td>CS31313 Microprocessors and Microcontrollers (Laboratory Course)</td>
<td>100</td>
</tr>
</tbody>
</table>
### 8.11 CS37401 Mini Project

### 8.12 CS30401 Comprehensive Viva Voce

### 8.13 CS37301 Seminar

### 8.14 CS37302 Project Stage - 1

### 9 Course Structure - Module VI

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS30102</td>
<td>Software Engineering (Theory Course)</td>
</tr>
<tr>
<td>CS30106</td>
<td>Database Management Systems (Theory Course)</td>
</tr>
<tr>
<td>CS30108</td>
<td>Design and Analysis of Algorithm (Theory Course)</td>
</tr>
<tr>
<td>CS30114</td>
<td>Systems Programming (Theory Course)</td>
</tr>
<tr>
<td>CS31119</td>
<td>Object Oriented Modeling and Design (Theory Course)</td>
</tr>
<tr>
<td>CS30102</td>
<td>Software Engineering (Tutorial)</td>
</tr>
<tr>
<td>CS30108</td>
<td>Design and Analysis of Algorithm (Tutorial)</td>
</tr>
<tr>
<td>CS30314</td>
<td>System programming (Laboratory Course)</td>
</tr>
<tr>
<td>CS31319</td>
<td>Object Oriented Modeling and Design (Laboratory Course)</td>
</tr>
<tr>
<td>CS30306</td>
<td>Database Management Systems (Laboratory Course)</td>
</tr>
<tr>
<td>CS37402</td>
<td>Mini Project</td>
</tr>
<tr>
<td></td>
<td>Professional Development Course (Institute Level)</td>
</tr>
<tr>
<td>CS30402</td>
<td>Comprehensive Viva Voce</td>
</tr>
<tr>
<td>CS37301</td>
<td>Seminar</td>
</tr>
</tbody>
</table>

### 10 Course Syllabi for Courses - Module VI

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS30102</td>
<td>Software Engineering (Theory Course)</td>
</tr>
<tr>
<td>CS30106</td>
<td>Database Management Systems (Theory Course)</td>
</tr>
<tr>
<td>CS30108</td>
<td>Design and Analysis of Algorithm (Theory Course)</td>
</tr>
<tr>
<td>CS30114</td>
<td>Systems Programming (Theory Course)</td>
</tr>
<tr>
<td>CS31119</td>
<td>Object Oriented Modeling and Design (Theory Course)</td>
</tr>
<tr>
<td>CS30102</td>
<td>Software Engineering (Tutorial)</td>
</tr>
<tr>
<td>CS30108</td>
<td>Design and Analysis of Algorithm (Tutorial)</td>
</tr>
<tr>
<td>CS30314</td>
<td>System programming (Laboratory Course)</td>
</tr>
<tr>
<td>CS31319</td>
<td>Object Oriented Modeling and Design (Laboratory Course)</td>
</tr>
<tr>
<td>CS30306</td>
<td>Database Management Systems (Laboratory Course)</td>
</tr>
<tr>
<td>CS37402</td>
<td>Mini Project</td>
</tr>
<tr>
<td>CS30402</td>
<td>Comprehensive Viva Voce</td>
</tr>
<tr>
<td>CS37301</td>
<td>Seminar</td>
</tr>
</tbody>
</table>

### 11 Course Structure - Module VII

### 12 Course Syllabi for Courses - Module VII

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40115</td>
<td>Distributed Computing (Theory Course)</td>
</tr>
<tr>
<td>CS40114</td>
<td>Business Intelligence and Analytics (Theory Course)</td>
</tr>
<tr>
<td>CS42105</td>
<td>Mobile Computing</td>
</tr>
<tr>
<td>CS42131</td>
<td>Enterprise Systems</td>
</tr>
<tr>
<td>CS42127</td>
<td>Cloud Computing</td>
</tr>
</tbody>
</table>

### Course Structure - Module VII

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40115</td>
<td>Distributed Computing (Theory Course)</td>
</tr>
<tr>
<td>CS40114</td>
<td>Business Intelligence and Analytics (Theory Course)</td>
</tr>
<tr>
<td>CS42105</td>
<td>Mobile Computing</td>
</tr>
<tr>
<td>CS42131</td>
<td>Enterprise Systems</td>
</tr>
<tr>
<td>CS42127</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>CS42103</td>
<td>Parallel Computing on GPU</td>
</tr>
<tr>
<td>CS42132</td>
<td>Network Security</td>
</tr>
<tr>
<td>CS42101</td>
<td>Advanced Computer Graphics</td>
</tr>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>CS42114</td>
<td>Product Design</td>
</tr>
<tr>
<td>CS42115</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>CS42134</td>
<td>Modeling and Simulation</td>
</tr>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>CS42114</td>
<td>Product Design</td>
</tr>
<tr>
<td>CS42115</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>CS42134</td>
<td>Modeling and Simulation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40115</td>
<td>Distributed Computing (Tutorial Course)</td>
<td>172</td>
</tr>
</tbody>
</table>

12.6  **Elective Group II (Tutorial Course)**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td>174</td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td>176</td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td>178</td>
</tr>
<tr>
<td>CS42114</td>
<td>Product Design</td>
<td>179</td>
</tr>
<tr>
<td>CS42115</td>
<td>Human Computer Interaction</td>
<td>180</td>
</tr>
<tr>
<td>CS42134</td>
<td>Modeling and Simulation</td>
<td>181</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40314</td>
<td>Business Intelligence and Analytics (Laboratory Course)</td>
<td>182</td>
</tr>
</tbody>
</table>

12.8  *Elective Group I (Laboratory Course)*

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42305</td>
<td>Mobile Computing</td>
<td>184</td>
</tr>
<tr>
<td>CS42331</td>
<td>Enterprise Systems</td>
<td>185</td>
</tr>
<tr>
<td>CS42327</td>
<td>Cloud Computing</td>
<td>187</td>
</tr>
<tr>
<td>CS42303</td>
<td>Parallel Computing on GPU</td>
<td>188</td>
</tr>
<tr>
<td>CS42332</td>
<td>Network Security</td>
<td>189</td>
</tr>
<tr>
<td>CS42301</td>
<td>Advanced Computer Graphics</td>
<td>191</td>
</tr>
</tbody>
</table>

12.9  CS47303  Project Stage - II  193

13  **Course Structure - Module VIII**

14  **Course Syllabi for Courses - Module VIII**

14.1 CS40106  Compiler Design (Theory Course)  198

14.2 CS40110  Artificial Intelligence (Theory Course)  200

14.3  $Elective Group III (Theory Course)

<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>202</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS42128</td>
<td>Machine Learning</td>
<td>204</td>
</tr>
<tr>
<td>CS42104</td>
<td>Neural Networks</td>
<td>206</td>
</tr>
<tr>
<td>CS42111</td>
<td>Software testing and Quality Assurance</td>
<td>208</td>
</tr>
<tr>
<td>CS42124</td>
<td>Algorithmic Number Theory and Algebra</td>
<td>211</td>
</tr>
</tbody>
</table>

14.4

**Elective Group IV (Theory Course)**

<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>214</td>
</tr>
<tr>
<td>CS42129</td>
<td>Management Information Systems</td>
<td>216</td>
</tr>
<tr>
<td>CS42121</td>
<td>Advanced Computer Architecture</td>
<td>219</td>
</tr>
<tr>
<td>CS42116</td>
<td>Convergence Technologies</td>
<td>222</td>
</tr>
<tr>
<td>CS42133</td>
<td>Embedded Systems</td>
<td>224</td>
</tr>
<tr>
<td>CS42130</td>
<td>Image Processing</td>
<td>226</td>
</tr>
</tbody>
</table>

14.5

**Elective Group III (Tutorial)**

<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>228</td>
</tr>
<tr>
<td>CS42128</td>
<td>Machine Learning</td>
<td>229</td>
</tr>
<tr>
<td>CS42104</td>
<td>Neural Networks</td>
<td>230</td>
</tr>
<tr>
<td>CS42111</td>
<td>Software testing and Quality Assurance</td>
<td>231</td>
</tr>
<tr>
<td>CS42124</td>
<td>Algorithmic Number Theory and Algebra</td>
<td>233</td>
</tr>
</tbody>
</table>

14.6

**Elective Group IV (Tutorial)**

<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>235</td>
</tr>
<tr>
<td>CS42129</td>
<td>Management Information Systems</td>
<td>236</td>
</tr>
<tr>
<td>CS42121</td>
<td>Advanced Computer Architecture</td>
<td>237</td>
</tr>
<tr>
<td>CS42116</td>
<td>Convergence Technologies</td>
<td>238</td>
</tr>
<tr>
<td>CS42130</td>
<td>Convergence Technologies</td>
<td>239</td>
</tr>
<tr>
<td>CS42133</td>
<td>Embedded Systems</td>
<td>241</td>
</tr>
</tbody>
</table>

14.7

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
</tbody>
</table>

14.8

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
</tbody>
</table>

14.9

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

15

**Course Syllabi for PD Courses in TY B.Tech (Computer Engineering)**

<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>250</td>
</tr>
<tr>
<td>CS33312</td>
<td>PIC Microcontroller</td>
<td>252</td>
</tr>
<tr>
<td>CS33313</td>
<td>Mobile Application Development</td>
<td>253</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>CS33306</td>
<td>Ethical Hacking and Network Defense</td>
<td>255</td>
</tr>
<tr>
<td>CS33310</td>
<td>Spring Framework</td>
<td>257</td>
</tr>
<tr>
<td>CS33314</td>
<td>Struts Framework</td>
<td>259</td>
</tr>
<tr>
<td>CS33311</td>
<td>Problem Solving and Programming</td>
<td>261</td>
</tr>
<tr>
<td>CS33315</td>
<td>Big Data Technologies</td>
<td>263</td>
</tr>
<tr>
<td>CS33307</td>
<td>Matlab</td>
<td>264</td>
</tr>
</tbody>
</table>
Program Educational Objectives (PEO)
B.Tech (Computer Engineering)
List of Programme Education Objectives [PEO] and Programme Outcomes [PO]

<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 as a committed technology workforce by providing them global educational platform with innovative practices resulting in computing artifacts realization</td>
</tr>
<tr>
<td>PEO2</td>
<td>Core competence</td>
<td>To impart adequate mathematical and computing theory knowledge basis leading to sustainable computer engineering solutions development</td>
</tr>
<tr>
<td>PEO3</td>
<td>Breadth</td>
<td>To inculcate problem solving skills and engineering practices in students adhering to well-formed technical specifications and constraints with the help of sound methods, tools and techniques</td>
</tr>
<tr>
<td>PEO4</td>
<td>Professionalism</td>
<td>To instill in the students professional and ethical practices by following effective guidelines to acquire aptitude, attitude and desire beneficial in societal context</td>
</tr>
<tr>
<td>PEO5</td>
<td>Learning Environment</td>
<td>To promote aspiring students for continuing education, engineering certifications and entrepreneurship in emerging areas of computing</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, mathematical and computing fundamentals in order to devise engineering solution for real world problems.  
2. To apply computer theory and algorithmic principles to innovatively craft solutions by context and development. |
| P02 | GA: 2 Problem Analysis | 3. To discover and infer computing problem situations, resulting in physical model, mathematical model or graphical model depicting the overall problem.  
4. To systematize functional specifications of target computing environment by adequate consideration of technology infrastructure, boundary conditions and constraints. |
| P03 | GA: 3: Design/Development of solution | 5. To conceive well-formed design specifications and constructs demonstrating correct compositional system structure with implementation-centric considerations.  
6. To incorporate architectural styles and design patterns to assimilate new facts, information and ideas about the design. |
| P04 | GA: 4: Conduct Investigation of Complex Problem | 7. To interpret reference data and program pragmatics for analyzable experimental results derivation.  
8. To judge and relate complexity issues and levels by making use of standardized verification and validation techniques. |
| P05 | GA: 5: Modern Tool Usage | 9. To operationalize and utilize the state-of-the-art CASE tools for engineering artifacts construction.  
10. To correlate and hypothesize problems for recognizing new or unfamiliar problem patterns. |
| P06 | GA: 6: The Engineer and Society | 11. To minimize adverse effects on the environment for their own and succeeding generations by respecting published facts and guidelines. |
| P07 | GA: 7: Environment and sustainability | 12. To consider the impact and benefits of engineering achievements in exploitation and management of technology on environment and society. |
| P08 | GA: 8: Ethics | 13. To prepare and present engineering evidence, theory and interpretations honestly, accurately and without bias. |
| P09 | GA: 9: Individual and Team Work | 14. To demonstrate high standards of professional conduct, openness and fairness by maintaining due respect towards rights and reputation of team members and development organization. |
| P10 | GA: 10: Communication | 15. To demonstrate deep listening, learning, leadership and managerial skills to solve complex engineering problems in teams. |
| P11 | GA: 11: Lifelong Learning | 16. To become part of a valuable body of knowledge in competitive computing areas.  
17. To acquire responsible positions in government, industry and society by continuously learning and researching. |
<table>
<thead>
<tr>
<th>PO12</th>
<th>GA: 12: Project Management and Finance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18. To creatively devise and incorporate project-specific processes supported by rigorous standards applicable to professional engineering bodies.</td>
</tr>
</tbody>
</table>
## F.Y. B. Tech. Structure with effect from Academic Year 2015-16

### 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</td>
<td>Tut.</td>
<td>ISA</td>
</tr>
<tr>
<td>HS10109</td>
<td>Linear Algebra and Random Variables</td>
<td>Theory - Core</td>
<td>3   -</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>HS10103</td>
<td>Modern Physics</td>
<td>Theory – Core</td>
<td>3   -</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>CH10101</td>
<td>Chemistry</td>
<td>Theory – Core</td>
<td>3   -</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>ME10101</td>
<td>Engineering Graphics</td>
<td>Theory – Core</td>
<td>3   -</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>HS16101</td>
<td>Sociology</td>
<td>Theory – OE</td>
<td>2   -</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>HS10301</td>
<td>Engineering Graphics Lab</td>
<td>Lab – Core</td>
<td>-   2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CH10301</td>
<td>Science Lab</td>
<td>Lab – Core</td>
<td>-   2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HS17401</td>
<td>Mini Project</td>
<td>Project</td>
<td>-   4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>14  08</td>
<td>2</td>
<td></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>3</td>
<td>-</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>HS10104</td>
<td>Engineering Mechanics</td>
<td>3</td>
<td>-</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>HS10108</td>
<td>Electrical Engineering Fundamentals</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>CS10102</td>
<td>Computer Programming</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>HS16102</td>
<td>Economics Management Technology</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>HS16104</td>
<td>Cost &amp; Acc. Business Law</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS16106</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS10302</td>
<td>Computer Programming</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HS10306</td>
<td>Engineering Lab</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HS17402</td>
<td>Mini Project</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>14</td>
<td>08</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### 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></td>
<td></td>
<td></td>
<td>Test 1</td>
<td>Test 2</td>
<td>HA</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></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>
### 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>HS17301</td>
<td>General seminar I</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>HS15306</td>
<td>French II</td>
<td>Language</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>HS15307</td>
<td>German II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS15308</td>
<td>Spanish II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS15309</td>
<td>Japanese II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS15310</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><strong>TOTAL</strong></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: (8+2 Hrs)


Unit 2: (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: (8+2 Hrs)


Part B: Preprocessor and preprocessor directives: macro substitution, difference between macro and functions.

Unit 4: (8+2 Hrs)
Part A:

Pointers: Concept of pointers, relevance of data type in pointer variable, pointer arithmetic.

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: (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 sting.  
   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 sting.  
   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 VI
### Module VI

<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>CS30102</td>
<td>Software Engineering</td>
<td>S1</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS30106</td>
<td>Database Management Systems</td>
<td>S2</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS30108</td>
<td>Design and Analysis of Algorithms</td>
<td>S3</td>
<td>3</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>CS31119</td>
<td>Object Oriented Modeling and Design</td>
<td>S4</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS30114</td>
<td>System Programming</td>
<td>S5</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS30314</td>
<td>System Programming</td>
<td>P1</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>CS30306</td>
<td>Database Management Systems</td>
<td>P2</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>CS30319</td>
<td>Object</td>
<td>P3</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Type</td>
<td>Credits</td>
<td>Contact</td>
<td>Total</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>-----------------</td>
<td>---------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>CS37402</td>
<td>Mini Project (T.Y. Semester I) Irrespective of Module</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30 70 2</td>
</tr>
<tr>
<td>CS37301</td>
<td>Seminar (T.Y. Semester I) Irrespective of Module</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30 70 1</td>
</tr>
<tr>
<td>PD</td>
<td>Institute Level Elective</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30 70 2</td>
</tr>
<tr>
<td>CS30402</td>
<td>Comprehensive Viva Voce</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30 70 2</td>
</tr>
<tr>
<td>CS37302</td>
<td>Project Stage 1 (T.Y. Semester II) Irrespective of Module</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30 70 2</td>
</tr>
</tbody>
</table>

**TOTAL** 14 16 2 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.

FF No. : 654 A

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)

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.
CS30108:: DESIGN AND ANALYSIS OF ALGORITHMS

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

Prerequisites: Data Structures

Unit 1: (9+1 Hrs)
Overview of Time Complexity analysis, Divide and Conquer
Part A: Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst case time and space complexity of algorithms. Overview of searching, sorting algorithms (binary search, insertion sort, heap sort, bubble sort). Adversary lower bounds (for comparison based sorting, for finding second minima etc). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity and to prove correctness of algorithms. Amortized complexity of algorithms.

Unit 2: (8+1 Hrs)
Dynamic Programming and Backtracking Strategies

Part A: Dynamic Programming: General strategy, simple dynamic programming based algorithms to compute Fibonacci numbers, binomial coefficients. Matrix Chain multiplication. Optimal binary search tree (OBST) construction, 0/1-Knapsack, Traveling Salesperson Problem, Shortest path in a Graph, Sequence Alignment problem, Scheduling problem.
Backtracking: General Strategy, n-Queen’s problem, Graph Coloring, Hamiltonian Cycles, 0/1 Knapsack, Subset sum problem.
Part B: String Editing Problem, Patience-sorting and O(n log n) algorithm for longest increasing sub-sequence problem. Solution for Peg-solitaire game.

Unit 3: (7+1 Hrs)
Greedy, Branch & Bound, Transform and Conquer techniques

Branch and Bound: General Strategy, 0/1 Knapsack, Traveling Salesperson Problem. Problem solving based on transform and conquer technique (Gaussian elimination, Horner’s rule and fast exponentiation etc). Heuristic based algorithms (Knight tour).
Part B: Postage stamp problem, n*n*n Queens problem, testing 2-colorability of graphs efficiently

Unit 4: (6+1 Hrs)
Introduction to Complexity Theory and NP-Completeness

**Part A:** Overview of deterministic and non deterministic Algorithms. Time Complexity classes P, NP, coNP, and their interrelation, EXP. Space complexity class PSPACE. Notion of polynomial time many one reductions reduction. Notion of NP-hardness and NP-completeness. Cook’s Theorem and implication to P versus NP question. NP-hardness of halting problem.


**Part B:** Decision Vs Search versions of problems in class NP, some problems in NP intersection coNP (linear programming, primality testing, perfect matchings in bipartite graphs).

**Unit 5:** (10+1 Hrs)

**Introduction to Randomized, approximation and online algorithms.**

**Part A: Randomized algorithms:** Introduction to Las-Vegas and Monte-Carlo Algorithms. Abundance of witnesses/solutions and application of randomization, solving SAT for formulas with “many” satisfying assignments, Randomized Quick Sort, Karger’s algorithm for Min Cut problem, Coupon Collector problem.

**Approximation algorithms:** Introduction to NP-optimization problems, factor-2 approximation algorithm for Vertex Cover, hardness of approximation of Travelling Sales Person Problem(TSP), factor-2 approximation algorithm for metric TSP, approximation algorithm for set-cover.

**Part B:** Birthday paradox, probabilistic recurrences, generation of large primes and Prime Number Theorem. Approximation algorithm for bin packing problem.

**Text Books**


**Reference Books**


**Course Outcomes:**

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

1. To analyze asymptotic time and space complexity of an algorithm for worst, average and best cases using suitable mathematical tools.
2. To formulate computational problems in mathematically precise manner
3. To design efficient algorithms for computational problems using appropriate algorithmic paradigm
4. To prove NP-completeness of some decision problems
5. To grasp the significance of the notion of NP-completeness and its relation with intractability of the decision problems
6. To explain the role of randomization and approximation in computation
CS30114:: SYSTEMS PROGRAMMING

Credits: 02  
Teaching Scheme: - Theory 2 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.  
(5+1 Hrs)

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: (6+1 Hrs)

**Encoding, Decoding and Device drivers**


**Part B:** Library Description for IA-32/Intel64.

Unit 5: (5+1 Hrs)

**TSR Programming**


**Text Books**


**Reference Books**


**Course Outcomes**

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

1. Discriminate among different System software and their functionalities.
2. Design Device Drivers, TSR programs and DLL for real world applications.
3. Interpret the methods and techniques about instructions Encoding and Decoding for implementing system-level programs.
4. Deliver the knowledge and techniques in order to bridge the gap between the society and technology.
5. Adapt the skills and ethics to solve critical problems about System design and provide solutions to real world problems.
6. Develop approaches and methods for implementing different system-level software's.
CS31119 : Object Oriented Modeling and Design

Credits: 03

Teaching Scheme: - Theory 3 Hrs/Week

Prerequisites: Data Structures

Unit 1: 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:
CS30108:: DESIGN AND ANALYSIS OF ALGORITHMS

Credits: 01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites: Data Structures and Files

List of Tutorials

1. Introduction to basic set theory, Mathematical Induction. Problem solving based on Induction and Recursion.
2. Formally proving correctness of algorithms using induction, loop invariants.
3. Studying asymptotic behavior of some non-standard functions like: \( H_n \), \( \log(n!) \), \( \log^*(n) \), Ackerman function etc. Problem solving based on asymptotic notations.
4. Problem solving based on simple binary-search like technique (perfect power testing, finding square roots efficiently etc.)
5. Introduction to Adversary lower bound technique and problem solving based on it (E.g. Adversary lower bound for finding two smallest elements in an array)
6. Problem solving based on Divide and Conquer technique
7. Divide and Conquer technique for problems in Computational Geometry (like convex hull computation, finding closest pair of points, discrete analog of Ham-Sandwich theorem etc)
8. Fast Fourier Transform and efficient uni-variate polynomial multiplication.
9. Problem solving based on Dynamic Programming strategy
10. Introduction to matroids and relation with Greedy strategy
11. Efficient implementation of Union-Find data structure and applications.
12. Problem solving based on Greedy strategy (formally proving optimality of solution for various greedy based problems)
13. Programming assignment on some problem based on Backtracking strategy (E.g. Sudoku solvers, Hi-Q solver, graph coloring etc.)
14. Relation between search and decision versions of problems in complexity class \( NP \) and their self reducibility property (particularly for \( SAT \), Graph-Isomorphism, Hamiltonian Cycle etc)
15. Interesting problems in the complexity class \( NP \) intersection co\( NP \), e.g. Linear Programming, Matchings in bipartite graphs, Primality testing.
16. Problem solving on Elementary Probability theory.
17. Problem solving on Elementary Probability theory.
18. Algorithms for enumeration and uniform generation of combinatorial objects (typically for permutations, subsets of fixed size, trees, derangements etc).
19. Problem solving based on design and analysis of approximation algorithms.
Text Books

Reference 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
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:
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 / corelated 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
CS37402::MINI PROJECT

Credits: 02

Guidelines:
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

Guidelines:
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.
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.
6. Cooperate with diverse Teams for delivering automation Solution.
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.
CS33306:: 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 -

1. Analyze Real world problems using Spring Framework Architecture, MVC
model, Aspect Oriented Programming (AOP) and Event Handling in Web Architecture.

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 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.
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 Hbas

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
CS33307:: MATLAB

Credits: 01

Teaching Scheme: - Laboratory 2 Hrs/Week

Prerequisites:

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

1. Solve Mathematical equations.
2. Design GUI by using MATLAB.
3. Construct Combinational circuit.
4. Validate design outputs using standards test equipments.
5. Develop animation programs by using MATLAB.
6. Perform various operations on Image.