## 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 (Computer Engineering)</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Course Structure - Module I and II</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>2.1 CS10102 Computer programming (Theory Course)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2.2 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></td>
<td>4.1 CS20117 Discrete Structures and Graph theory (Theory Course)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>4.2 CS20111 Data Structures (Theory Course)</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>4.3 CS20108 Computer Organization (Theory Course)</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>4.4 CS20116 Problem Solving and Programming (Theory Course)</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>4.5 CS20113 Digital Electronics and Logic Design (Theory Course)</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>4.6 CS20216 Problem Solving and Programming (Tutorial)</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>4.7 CS20113 Digital Electronics and Logic Design (Tutorial)</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>4.8 CS20311 Data structures (Laboratory Course)</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>4.9 CS20313 Digital Electronics (Laboratory Course)</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>4.10 CS27401 Mini Project</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>4.11 CS24306 PHP MYSQL OR CS24303 C#.NET (Skill Development Course)</td>
<td>43</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>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></td>
<td>6.1 CS21104 Mathematical Transformations and Applications (Theory Course)</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>Course Code</td>
<td>Course Name</td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>6.2</td>
<td>CS20110</td>
<td>Computer Graphics (Theory Course)</td>
</tr>
<tr>
<td>6.3</td>
<td>CS20105</td>
<td>Principles of Programming Languages (Theory Course)</td>
</tr>
<tr>
<td>6.4</td>
<td>CS21112</td>
<td>Data Communication (Theory Course)</td>
</tr>
<tr>
<td>6.5</td>
<td>CS20114</td>
<td>Microprocessor and Interfacing (Theory Course)</td>
</tr>
<tr>
<td>6.6</td>
<td>CS20114</td>
<td>Microprocessor and Interfacing (Tutorial)</td>
</tr>
<tr>
<td>6.7</td>
<td>CS21104</td>
<td>Mathematical Transforms and Applications (Tutorial)</td>
</tr>
<tr>
<td>6.8</td>
<td>CS20314</td>
<td>Microprocessor and Interfacing (Laboratory Course)</td>
</tr>
<tr>
<td>6.9</td>
<td>CS20305</td>
<td>Principles of Programming Languages (Laboratory Course)</td>
</tr>
<tr>
<td>6.10</td>
<td>CS20310</td>
<td>Computer Graphics (Laboratory Course)</td>
</tr>
<tr>
<td>6.11</td>
<td>CS27402</td>
<td>Mini Project</td>
</tr>
<tr>
<td>6.12</td>
<td>CS24302</td>
<td>ASP.NET OR</td>
</tr>
<tr>
<td></td>
<td>CS24304</td>
<td>Python (Skill Development Course)</td>
</tr>
<tr>
<td></td>
<td>CS24307</td>
<td>Ruby Programming</td>
</tr>
<tr>
<td>6.13</td>
<td></td>
<td>Elective – Health &amp; Hobby</td>
</tr>
<tr>
<td>6.14</td>
<td>CS20402</td>
<td>Comprehensive Viva Voce</td>
</tr>
</tbody>
</table>

### 7 Course Structure - Module V

### 8 Course Syllabi for Courses - Module V

<table>
<thead>
<tr>
<th></th>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>CS30101</td>
<td>Operating Systems (Theory Course)</td>
<td>80</td>
</tr>
<tr>
<td>8.2</td>
<td>CS30116</td>
<td>Computer Networks (Theory Course)</td>
<td>83</td>
</tr>
<tr>
<td>8.3</td>
<td>CS30105</td>
<td>Theory of Computation (Theory Course)</td>
<td>86</td>
</tr>
<tr>
<td>8.4</td>
<td>CS31113</td>
<td>Microprocessors and Microcontrollers (Theory Course)</td>
<td>89</td>
</tr>
<tr>
<td>8.5</td>
<td>CS31115</td>
<td>Advanced Data Structures (Theory Course)</td>
<td>91</td>
</tr>
<tr>
<td>8.6</td>
<td>CS30101</td>
<td>Operating Systems (Tutorial)</td>
<td>93</td>
</tr>
<tr>
<td>8.7</td>
<td>CS31215</td>
<td>Advanced Data Structures (Tutorial)</td>
<td>95</td>
</tr>
<tr>
<td>8.8</td>
<td>CS30303</td>
<td>Operating Systems (Laboratory Course)</td>
<td>96</td>
</tr>
<tr>
<td>8.9</td>
<td>CS30316</td>
<td>Computer Networks (Laboratory Course)</td>
<td>98</td>
</tr>
<tr>
<td>8.10</td>
<td>CS31313</td>
<td>Microprocessors and Microcontrollers (Laboratory Course)</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Course Code</td>
<td>Course Title</td>
<td>Module</td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>8.11</td>
<td>CS37401</td>
<td>Mini Project</td>
<td>101</td>
</tr>
<tr>
<td>8.12</td>
<td></td>
<td>Professional Development Course (Institute Level)</td>
<td>-</td>
</tr>
<tr>
<td>8.13</td>
<td>CS30401</td>
<td>Comprehensive Viva Voce</td>
<td>-</td>
</tr>
<tr>
<td>8.14</td>
<td>CS37301</td>
<td>Seminar</td>
<td>102</td>
</tr>
<tr>
<td>8.15</td>
<td>CS37302</td>
<td>Project Stage - 1</td>
<td>103</td>
</tr>
</tbody>
</table>

9 Course Structure - Module VI

<table>
<thead>
<tr>
<th></th>
<th>Course Code</th>
<th>Course Title</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1</td>
<td>CS30102</td>
<td>Software Engineering (Theory Course)</td>
<td>109</td>
</tr>
<tr>
<td>10.2</td>
<td>CS30106</td>
<td>Database Management Systems (Theory Course)</td>
<td>111</td>
</tr>
<tr>
<td>10.3</td>
<td>CS30108</td>
<td>Design and Analysis of Algorithm (Theory Course)</td>
<td>113</td>
</tr>
<tr>
<td>10.4</td>
<td>CS30114</td>
<td>Systems Programming (Theory Course)</td>
<td>116</td>
</tr>
<tr>
<td>10.5</td>
<td>CS31119</td>
<td>Object Oriented Modeling and Design (Theory Course)</td>
<td>118</td>
</tr>
<tr>
<td>10.6</td>
<td>CS30102</td>
<td>Software Engineering (Tutorial)</td>
<td>121</td>
</tr>
<tr>
<td>10.7</td>
<td>CS30108</td>
<td>Design and Analysis of Algorithm (Tutorial)</td>
<td>122</td>
</tr>
<tr>
<td>10.8</td>
<td>CS30314</td>
<td>System programming (Laboratory Course)</td>
<td>124</td>
</tr>
<tr>
<td>10.9</td>
<td>CS31319</td>
<td>Object Oriented Modeling and Design (Laboratory Course)</td>
<td>126</td>
</tr>
<tr>
<td>10.10</td>
<td>CS30306</td>
<td>Database Management Systems (Laboratory Course)</td>
<td>128</td>
</tr>
<tr>
<td>10.11</td>
<td>CS37402</td>
<td>Mini Project</td>
<td>130</td>
</tr>
<tr>
<td>10.12</td>
<td></td>
<td>Professional Development Course (Institute Level)</td>
<td>-</td>
</tr>
<tr>
<td>10.13</td>
<td>CS30402</td>
<td>Comprehensive Viva Voce</td>
<td>-</td>
</tr>
<tr>
<td>10.14</td>
<td>CS37301</td>
<td>Seminar</td>
<td>131</td>
</tr>
</tbody>
</table>

11 Course Structure - Module VII

<table>
<thead>
<tr>
<th></th>
<th>Course Code</th>
<th>Course Title</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1</td>
<td>CS40115</td>
<td>Distributed Computing (Theory Course)</td>
<td>135</td>
</tr>
<tr>
<td>12.2</td>
<td>CS40114</td>
<td>Business Intelligence and Analytics (Theory Course)</td>
<td>138</td>
</tr>
<tr>
<td>12.3</td>
<td></td>
<td>*Elective Group I (Theory Course)</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>CS42105</td>
<td>• Mobile Computing</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>CS42131</td>
<td>• Enterprise Systems</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>CS42127</td>
<td>• Cloud Computing</td>
<td>147</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Credits</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>CS42103</td>
<td>Parallel Computing on GPU</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>CS42132</td>
<td>Network Security</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>CS42101</td>
<td>Advanced Computer Graphics</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td>157</td>
<td></td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td>163</td>
<td></td>
</tr>
<tr>
<td>CS42114</td>
<td>Product Design</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>CS42115</td>
<td>Human Computer Interaction</td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>CS42134</td>
<td>Modeling and Simulation</td>
<td>170</td>
<td></td>
</tr>
<tr>
<td>CS40115</td>
<td>Distributed Computing (Tutorial Course)</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td>CS42125</td>
<td>Randomized and Approximation Algorithms</td>
<td>174</td>
<td></td>
</tr>
<tr>
<td>CS42113</td>
<td>Digital Signal Processing</td>
<td>176</td>
<td></td>
</tr>
<tr>
<td>CS42119</td>
<td>Information Retrieval</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>CS42114</td>
<td>Product Design</td>
<td>179</td>
<td></td>
</tr>
<tr>
<td>CS42115</td>
<td>Human Computer Interaction</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>CS42134</td>
<td>Modeling and Simulation</td>
<td>181</td>
<td></td>
</tr>
<tr>
<td>CS40314</td>
<td>Business Intelligence and Analytics (Laboratory Course)</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>CS42305</td>
<td>Mobile Computing</td>
<td>184</td>
<td></td>
</tr>
<tr>
<td>CS42331</td>
<td>Enterprise Systems</td>
<td>185</td>
<td></td>
</tr>
<tr>
<td>CS42327</td>
<td>Cloud Computing</td>
<td>187</td>
<td></td>
</tr>
<tr>
<td>CS42303</td>
<td>Parallel Computing on GPU</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td>CS42332</td>
<td>Network Security</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>CS42301</td>
<td>Advanced Computer Graphics</td>
<td>191</td>
<td></td>
</tr>
<tr>
<td>CS47303</td>
<td>Project Stage - II</td>
<td>193</td>
<td></td>
</tr>
</tbody>
</table>

13 Course Structure - Module VIII

14 Course Syllabi for Courses - Module VIII

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40106</td>
<td>Compiler Design (Theory Course)</td>
<td>198</td>
</tr>
<tr>
<td>CS40110</td>
<td>Artificial Intelligence (Theory Course)</td>
<td>200</td>
</tr>
<tr>
<td>CS42120</td>
<td>Data Mining</td>
<td>202</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>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 Name</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 Name</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 Name</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>CS42133</td>
<td>Embedded Systems</td>
<td>239</td>
</tr>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

14.7

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

14.8

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

14.9

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

14.10

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

14.11

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<tr>
<td>CS47308</td>
<td>Artificial Intelligence (Laboratory Course)</td>
<td>245</td>
</tr>
</tbody>
</table>

Project Stage - III

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS40306</td>
<td>Image Processing</td>
<td>241</td>
</tr>
<tr>
<td>CS40310</td>
<td>Compiler Design (Laboratory Course)</td>
<td>243</td>
</tr>
<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 Name</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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>15.4</td>
<td>CS33306</td>
<td>Ethical Hacking and Network Defense</td>
</tr>
<tr>
<td>15.5</td>
<td>CS33310</td>
<td>Spring Framework</td>
</tr>
<tr>
<td>15.6</td>
<td>CS33314</td>
<td>Struts Framework</td>
</tr>
<tr>
<td>15.7</td>
<td>CS33311</td>
<td>Problem Solving and Programming</td>
</tr>
<tr>
<td>15.8</td>
<td>CS33315</td>
<td>Big Data Technologies</td>
</tr>
<tr>
<td>15.9</td>
<td>CS33307</td>
<td>Matlab</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>
Course Name: Computer Programming

Course Code: CS101  Course Type: THL

Credits: 5  Teaching Scheme:  Theory: 3 Hours / Week
Lab/Tut.: 4 Hours / Week

Unit 1: Introduction to programming  (8 Hours)
Problem solving using computers; algorithms and flowcharts; using simplecpp graphics
commands; notion of syntax and semantics; Repeating a block of commands; Nested
repeat; numerical functions; comments; Data types; identifiers; initialization; const; Input
and Output; Arithmetic operators; programming idioms; Compound assignment; blocks;
scope and shadowing.

Unit 2: Flow of Control  (7 Hours)
Conditional Constructs: Relational and Logical Operators, various forms of if..else
statements, ternary operator, switch..case statement;
Loops: Types of Loops, while, do..while, for, break and continue, goto statement.

Unit 3: Array, Strings and Pointers  (10 Hours)
Arrays: Definition, syntax, element operations, memory representation, initialization.
Two dimensional array, row and column major;
Applications of arrays: sorting, searching and matrix operations;
Pointers: Definition, syntax, address of operator, pointer variables, relevance of data
type in pointer variables, dereferencing operator, Pointer to pointer, address arithmetic;
Array and Pointers: Accessing array elements using pointers;
Strings: Introduction, Array of characters, output, input, character string constant,
Accepting multiword string, Array of strings.

Unit 4: Functions and Recursion  (7 Hours)
Functions: Introduction; definition; anatomy of function; execution of function; Scope of
various types of variable; scope; local and global variables; nested function call; returning
values from function; Menu driven programs;
Function and Pointers: Call by value v/s call by address, Passing array to function,
returning pointers from function, dangling pointers;
Function and strings: Library functions from string.h library, building user defined
functions for string operations;
Recursive Function: Definition, Examples, Types of recursion.
Unit 5: Structures (6 Hours)

Structure: Need, Definition, syntax, declaration and initialization, structure variables, accessing and assigning values to structure variables, “dot” operator;

Applications of structure: Functions and structures, array and structure, structure within a structure, pointers and structures.

Unit 6: File handling (4 Hours)

Introduction to file handling, using file pointers, file opening modes, reading from file, writing into file, closing file.
List of Practical:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Graphics: Drawing basic diagrams using SimpleCpp</td>
</tr>
<tr>
<td>2</td>
<td>Making computer calculate stuff : calculating average, finding roots of quadratic equation, complex numbers etc.</td>
</tr>
<tr>
<td>3</td>
<td>Switch statement: controlling the turtle</td>
</tr>
<tr>
<td>4</td>
<td>Loops, Conditionals and Arithmetic operations: Converting base of a number, Generating arithmetic and geometric progressions, Buttons on Canvas, Reversing n digit number etc.</td>
</tr>
<tr>
<td>5</td>
<td>Example: marks display program, find min/max Generate roll numbers</td>
</tr>
<tr>
<td>6</td>
<td>Linear and Binary Search</td>
</tr>
<tr>
<td>7</td>
<td>Bubble sort</td>
</tr>
<tr>
<td>8</td>
<td>Insertion and Selection Sort</td>
</tr>
<tr>
<td>9</td>
<td>Matrix arithmetic</td>
</tr>
<tr>
<td>10</td>
<td>Demonstrating pointer to pointer, pointer arithmetic</td>
</tr>
<tr>
<td>11</td>
<td>Assignment on functions – menu driven programs, etc.</td>
</tr>
<tr>
<td>12</td>
<td>Using library functions</td>
</tr>
<tr>
<td>13</td>
<td>Implementing User defined string functions</td>
</tr>
<tr>
<td>14</td>
<td>Calculate factorial, generating progressions, GCD, LCM using recursion</td>
</tr>
<tr>
<td>15</td>
<td>Create Structures for: student result, employee payroll, library book issuing</td>
</tr>
<tr>
<td>16</td>
<td>File handling: Store employee records in a file</td>
</tr>
</tbody>
</table>

Text Books:

Reference Books:

Course Outcomes:
Upon completion of the course, graduates will be able to –
1. List procedural programming benefits to solve real world problems using generic prototype
2. Develop visual representation of problem in hand.
3. Apply available algorithmic principles to generate efficient solutions.
4. Justify modular programming approach by making use of elementary as well as superior data structures.
5. Apply programming fundamentals to solve real world problems using generic prototype.
6. Evaluate and manipulate given solutions in reengineered view.
Course Name: Software Workshop

Course Code: CS102  Course Type: LTH

Credits: 2  Teaching Scheme:  Theory: 1 Hours / Week
Lab/Tut.: 2 Hours / Week

Theory:

Unit 1: Installation and Configuration of Operating system  (2 Hours)
Introduction to Linux operating system; basic commands of Unix; Disk Partitioning

Unit 2: Software Installations  (3 Hours)
Types of software: Application software, system software (device drivers, etc); Levels of
programming language: Machine, Assembly, High level language; Software evolution:
Procedure oriented, Object oriented, Rule Based, Applicative language (Event driven);
Compilation model (Linker, loader, compiler, interpreter); Types of Virus and role of
anti-virus software; plagiarism checker; Patching; Key-terms: Open source, free and paid
software

Unit 3: System Administration  (1 Hours)
Account Management; Introduction to Control Panel and Admin Tools; Computer
Management GUI tool

Unit 4: Disk Administration  (1 Hours)
Disk Formatting; Disk Defragmentation; File and directory layout; File permissions

Unit 5: Introduction to MATLAB  (2 Hours)
Introduction; application development; working environment; handling graphics; libraries
and tool boxes; Introduction to Simulink

Unit 6: Introduction to SCILAB  (1 Hours)
Introduction; application development; working environment; equivalent operations

Unit 7: Introduction to Circuit Simulation Software  (2 Hours)
Introduction; circuit development; working environment; Simulation of simple circuits

Unit 8: Introduction to LaTeX  (2 Hours)
Introduction; Basic commands; writing mathematical equations; Scripting Tags
List of Practical:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Installation and Configuration of Operating system</td>
</tr>
<tr>
<td></td>
<td>a. Disk partitioning (Dual boot)</td>
</tr>
<tr>
<td></td>
<td>b. Linux operating system installation and configuration</td>
</tr>
<tr>
<td></td>
<td>c. Basic unix commands</td>
</tr>
<tr>
<td>2</td>
<td>Software Installations</td>
</tr>
<tr>
<td></td>
<td>a. Device drivers</td>
</tr>
<tr>
<td></td>
<td>b. Anti-virus software</td>
</tr>
<tr>
<td></td>
<td>c. Patching</td>
</tr>
<tr>
<td>3</td>
<td>System Administration</td>
</tr>
<tr>
<td></td>
<td>a. Account Management</td>
</tr>
<tr>
<td></td>
<td>b. Control Panel and Admin Tools</td>
</tr>
<tr>
<td></td>
<td>c. Computer Management GUI tool</td>
</tr>
<tr>
<td>4</td>
<td>Disk Administration</td>
</tr>
<tr>
<td></td>
<td>a. Disk Formatting</td>
</tr>
<tr>
<td></td>
<td>b. Disk Defragmentation</td>
</tr>
<tr>
<td></td>
<td>c. File and directory layout</td>
</tr>
<tr>
<td></td>
<td>d. File permissions</td>
</tr>
<tr>
<td>5</td>
<td>Introduction to MATLAB</td>
</tr>
<tr>
<td></td>
<td>a. Installation</td>
</tr>
<tr>
<td></td>
<td>b. Basic commands</td>
</tr>
<tr>
<td></td>
<td>c. Predefined libraries</td>
</tr>
<tr>
<td></td>
<td>d. Introduction to Simulink</td>
</tr>
<tr>
<td>6</td>
<td>Introduction to SCILAB</td>
</tr>
<tr>
<td></td>
<td>a. Installation</td>
</tr>
<tr>
<td></td>
<td>b. Basic commands</td>
</tr>
<tr>
<td></td>
<td>c. Predefined libraries</td>
</tr>
<tr>
<td>7</td>
<td>Introduction to Circuit Simulation Software</td>
</tr>
<tr>
<td></td>
<td>a. Installation</td>
</tr>
<tr>
<td></td>
<td>b. Simulation of simple circuits</td>
</tr>
<tr>
<td>8</td>
<td>Introduction to LaTeX</td>
</tr>
<tr>
<td></td>
<td>a. Installation</td>
</tr>
<tr>
<td></td>
<td>b. Basic commands</td>
</tr>
<tr>
<td></td>
<td>c. Writing mathematical equations</td>
</tr>
<tr>
<td></td>
<td>d. Scripting Tags</td>
</tr>
</tbody>
</table>

Text Books:


Reference Books:

Course Outcomes:
At the end of this course the student will be able to:
1. Understand basics of operating system and software evolution.
2. Know system administration
3. Write simple programs for data processing
4. Use software for simulation of electronic circuits
5. Organize documents effectively
MODULE III
## S.Y. B. Tech. Structure with effect from Academic Year 2015-16

### Module III

<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>ISA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tut.</td>
<td></td>
<td>Test 1</td>
<td>Test 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HA</td>
<td>Tut.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CA</td>
<td>ESE</td>
</tr>
<tr>
<td>CS20117</td>
<td>Discrete Structures and Graph Theory</td>
<td>$S_1$</td>
<td>3</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>CS20111</td>
<td>Data Structures</td>
<td>$S_2$</td>
<td>3 4</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>CS20108</td>
<td>Computer Organization</td>
<td>$S_3$</td>
<td>3</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>CS20116</td>
<td>Problem Solving and Programming</td>
<td>$S_4$</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>*CS20113</td>
<td>Digital Electronics and Logic Design Microprocessor and Interfacing</td>
<td>$S_5$</td>
<td>3 2 1</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>#CS20114</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS20311</td>
<td>Data Structures</td>
<td>$P_1$</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>*CS20313</td>
<td>Digital Electronics Microprocessor and Interfacing</td>
<td>$P_2$</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#CS20314</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS27402</td>
<td>Mini Project</td>
<td>$MP_3$</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS24306</td>
<td>PHP MYSQL OR C#.NET</td>
<td>$SD_3$</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS24303</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*LAB_3</td>
<td>Technical Writing</td>
<td>$LAB_3$</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

* CS24303
* CS24306

<table>
<thead>
<tr>
<th>#LAB₃</th>
<th>General Seminar – 2</th>
<th>LAB₃</th>
<th></th>
<th></th>
<th>70</th>
<th>30</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS20401</td>
<td>Comprehensive Viva Voce</td>
<td>CVV₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>16</td>
<td>14</td>
<td>2</td>
<td></td>
<td>25</td>
</tr>
</tbody>
</table>

# Students will register only in Semester IV irrespective of Module * Students will register only in Semester III irrespective of Module
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 about 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 via the formal language of propositional and predicate logic
6) **Demonstrate** use of pigeon-hole and inclusion-exclusion principle in solving elegant and important problems
CS20111:: Data Structures

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

Unit 1: Fundamentals of Data Structures, Sorting & Searching (8 Hrs)
Part A
Part B: External Sorting, Sparse Matrix: Addition and Fast transpose

Unit 2: Elementary Data Structures (8 Hrs)
Part A
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. Handle operations like searching, insertion, deletion, traversing mechanism etc. on data structures like stack and Queue.
3. Use linear and nonlinear data structures like stacks, queues, linked list etc
4. Demonstrate the use of tree and its applications.
5. Analyze the real world problems using Graph Data Structure.
6. Apply an appropriate data structure and algorithm to solve a problem
CS20108:: COMPUTER ORGANIZATION

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

Prerequisites:

Unit 1: 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 -

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operationalize arithmetic and control unit based on computer architectures.</td>
</tr>
<tr>
<td>2</td>
<td>Analyze impact of circuit driven and program driven analogies to assemble realizable solutions.</td>
</tr>
<tr>
<td>3</td>
<td>Design processor and memory with due consideration to tradeoffs and performance issues.</td>
</tr>
<tr>
<td>4</td>
<td>Suggest performance bound solutions in order to demonstrate variety of technologies.</td>
</tr>
<tr>
<td>5</td>
<td>Recognize historical scenario of computing unit’s development with regards to technological evolution.</td>
</tr>
<tr>
<td>6</td>
<td>Illustrate organization of digital computers with basic principles and operations of its components.</td>
</tr>
</tbody>
</table>
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 A.


Part B. Simpson’s rule, Trapezoidal rule

Unit 4: Recursion and Arrays  (6+1 Hrs)

Part A.

**Problem solving using recursion** - Virahanka numbers, Game of Nim. Sample generation, combination generation, permutation generation. Structural recursion: maintaining an ordered set, generating trees and layout of mathematical formulae.

**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 -

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apply logical ability to solve the problems using suitable technique.</td>
</tr>
<tr>
<td>2</td>
<td>Construct various algorithms based on different data types using fundamental and factoring methods</td>
</tr>
<tr>
<td>3</td>
<td>Develop programming methods to solve various numerical and differential equations.</td>
</tr>
<tr>
<td>4</td>
<td>Solve problems using recursive and iterative techniques.</td>
</tr>
<tr>
<td>5</td>
<td>Implement robust programming solutions for problems such as string processing and pattern searching</td>
</tr>
<tr>
<td>6</td>
<td>Select appropriate programming paradigm, data structures and algorithm to solve complex computing problem.</td>
</tr>
</tbody>
</table>
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

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 -

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Optimize logical equations using reduction techniques</td>
</tr>
<tr>
<td>2</td>
<td>Design different types of code convertors</td>
</tr>
<tr>
<td>3</td>
<td>Construct Combinational and Sequential circuits</td>
</tr>
<tr>
<td>4</td>
<td>Validate the internal structure of combinational circuits</td>
</tr>
<tr>
<td>5</td>
<td>Develop applications of sequential circuits</td>
</tr>
<tr>
<td>6</td>
<td>Describe Programmable Logic Devices</td>
</tr>
</tbody>
</table>
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          FF No. : 654 C

Credits: -01

Teaching Scheme: - Tutorial 1 Hr/Week

Prerequisites:

List of Contents

1. Introduction to electronic workbench, solving 2-3 combinational examples.
2. Introduction to 74XX, 54XX ICs and basic component of logic family
3. Diode as a switch.
4. Design logic gates using diodes.
5. Introduction to basic component of logic family-Transistor.
6. Characteristics and configuration of Transistor.
7. Transistor as a switch
8. Design logic gates using Transistor
9. Introduction to basic component of logic family-MOSFET
10. Study of voltage and current characteristics of digital ICs.
11. Design of state transition diagram using ASM.
12. Design of datapath and control path using RTL. (Two examples)

Text Books


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.
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

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.
CS24306 PHP MYSQL

Credits: 01

Teaching Scheme: Lab 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.