CURRICULUM
for the Academic year 2018 – 2019
Batch (2018-2020)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

I - IV Semester M. Tech (CSE)

COMPUTER SCIENCE AND ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE – 54
About the Institute

Ramaiah Institute of Technology (RIT) (formerly known as M. S. Ramaiah Institute of Technology) is a self-financing institution established in Bangalore in the year 1962 by the industrialist and philanthropist, Late Dr. M S Ramaiah. The institute is accredited with A grade by NAAC in 2016 and all engineering departments offering bachelor degree programs have been accredited by NBA. RIT is one of the few institutes with faculty student ratio of 1:15 and achieves excellent academic results. The institute is a participant of the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. All the departments are full with competent faculty, with 100% of them being postgraduates or doctorates. Some of the distinguished features of RIT are: State of the art laboratories, individual computing facility to all faculty members. All research departments are active with sponsored projects and more than 130 scholars are pursuing PhD. The Centre for Advanced Training and Continuing Education (CATCE), and Entrepreneurship Development Cell (EDC) have been set up on campus. RIT has a strong Placement and Training department with a committed team, a fully equipped Sports department, large air-conditioned library with over 80,000 books with subscription to more than 300 International and National Journals. The Digital Library subscribes to several online e-journals like IEEE, JET etc. RIT is a member of DELNET, and AICTE INDEST Consortium. RIT has a modern auditorium, several hi-tech conference halls, all air-conditioned with video conferencing facilities. It has excellent hostel facilities for boys and girls. RIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association. RIT obtained Academic Autonomy for all its UG and PG programs in the year 2007. As per the National Institutional Ranking Framework, MHRD, Government of India, Ramaiah Institute of Technology has achieved 45th rank in 2017 among the top 100 engineering colleges across India and occupied No. 1 position in Karnataka, among the colleges affiliated to VTU, Belagavi.
About the Department

<table>
<thead>
<tr>
<th>Year of Establishment</th>
<th>1984</th>
</tr>
</thead>
</table>
| Names of the Programmes offered | 1. UG: B.E. in Computer Science and Engineering  
2. PG: M.Tech. in Computer Science and Engineering  
3. PG: M.Tech. in Computer Networks and Engineering  
4. Ph.D  
5. M.Sc(Engg.) by Research |

The Department of Computer Science and Engineering (CSE) has eminent emeritus professors, 10 faculty with the doctorate degree and 15 pursuing the doctoral studies. The faculty has been publishing research papers in refereed journals and in conference proceedings. The department also conducts vocational courses and proficiency courses on fundamental and new programming languages and computer science concepts. These courses are conducted beyond college hours/summer semester by the faculty of the department. Some of the faculty are involved in institutional level activities and actively involved in interdisciplinary research activities. The department has state of the art laboratories like SAP, IBM Centre of Excellence and CUDA learning center. Technical seminars, workshops and hackathons are conducted regularly for UG & PG students. The department encourages the students to conduct and participate in extra-curricular/sports activities. The alumni network is very active and regular meeting are conducted by the department. The department is accredited by Nation Board of Accreditation (NBA). The department has MoUs with leading IT Industries like NVIDIA, SAP, IBM and HP. The department conducts subjects with more of hands-on sessions and encourages students to take up MOOC based online courses in NPTEL, IITBombayX, Coursera, Udacity and edX.
VISION OF THE INSTITUTE
To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio economic needs

MISSION OF THE INSTITUTE
MSRIT shall meet the global socio-economic needs through
  • Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
  • Establishing research clusters in emerging areas in collaboration with globally reputed organizations
  • Establishing innovative skills development, techno-entrepreneurial activities and consultancy for socio-economic needs

QUALITY POLICY
We at M. S. Ramaiah Institute of Technology strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management System complemented by the synergistic interaction of the stake holders concerned

VISION OF THE DEPARTMENT
To build a strong learning and research environment in the field of Computer Science and Engineering that responds to the challenges of 21st century.

MISSION OF THE DEPARTMENT
  • To produce computer science graduates who, trained in design and implementation of computational systems through competitive curriculum and research in collaboration with industry and other organizations
  • To educate students in technology competencies by providing professionally committed faculty and staff
  • To inculcate strong ethical values, leadership abilities and research capabilities in the minds of students so as to work towards the progress of the society
PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

An M.Tech (Computer Science & Engineering) graduate of M S Ramaiah Institute of Technology should, within three to five years of graduation

PEO1 Pursue a successful career in the field of Computer Science & Engineering or a related field utilizing his/her education and contribute to the profession as an excellent employee, or as an entrepreneur

PEO2 Be aware of the developments in the field of Computer Science & Engineering, continuously enhance their knowledge informally or by pursuing doctoral studies and engage in research and inquiry leading to new innovations and products

PEO3 Be able to work effectively in multidisciplinary and multicultural environments and be responsible members and leaders of their communities

PEO4 Understand the human, social and environmental context of their profession and contribute positively to the needs of individuals and society at large

PROGRAM OUTCOMES (POs):

PO1: An ability to independently carry out research/investigation and development work to solve practical problems

PO2: An ability to write and present a substantial technical report/document

PO3: Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program.

PO4: Acquire professional and intellectual integrity to stress upon the impact of computer engineering applications with respect to economic and environmental aspects

PO5: Acquire methods of engaging in life-long learning not only to predict and plan the projects of the future but also to groom others in the group.
<table>
<thead>
<tr>
<th>Semester</th>
<th>Humanities &amp; Social Sciences (HSS)</th>
<th>Basic Sciences/ Lab (BS)</th>
<th>Engineering Sciences/ Lab (ES)</th>
<th>Professional Courses - Core (Hard core, soft core, Lab) (PC-C)</th>
<th>Professional Courses-Electives (PC-E)</th>
<th>Other Electives (OE)</th>
<th>Project Work/ Internship (PW /IN)</th>
<th>Extra &amp; Co-curricular activities (EAC)</th>
<th>Total credits in a semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>Second</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>Third</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>Fourth</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>88</td>
</tr>
</tbody>
</table>
# SCHEME OF TEACHING
## I SEMESTER

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Name</th>
<th>Category</th>
<th>Credits</th>
<th>Contact Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>1.</td>
<td>MCS11</td>
<td>Advanced Engineering Mathematics</td>
<td>PC-C</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2.</td>
<td>MCS12</td>
<td>Advances in Database Management Systems</td>
<td>PC-C</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>MCSExx</td>
<td>Elective-I</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>4.</td>
<td>MCSExx</td>
<td>Elective-II</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>5.</td>
<td>MCSExx</td>
<td>Elective-III</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>6.</td>
<td>MCSL13</td>
<td>Software Development for Portable Devices Laboratory</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7.</td>
<td>MCSL14</td>
<td>Python Programming Laboratory</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8.</td>
<td>MCS15</td>
<td>Technical Seminar-I</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## II SEMESTER

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Name</th>
<th>Category</th>
<th>Credits</th>
<th>Contact Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>1.</td>
<td>MCS21</td>
<td>Data Analytics Using R</td>
<td>PC-C</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2.</td>
<td>MCS22</td>
<td>Advanced Algorithms</td>
<td>PC-C</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3.</td>
<td>MCSExx</td>
<td>Elective-IV</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>4.</td>
<td>MCSExx</td>
<td>Elective-V</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>5.</td>
<td>MCSExx</td>
<td>Elective-VI</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>6.</td>
<td>MCSL23</td>
<td>IoT and Cloud Computing Laboratory</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7.</td>
<td>MCSL24</td>
<td>Data Analytics Lab</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8.</td>
<td>MCS25</td>
<td>Technical Seminar-II</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Total:** 24
### III SEMESTER

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Name</th>
<th>Category</th>
<th>Credits</th>
<th>Contact Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>MCS31</td>
<td>Engineering Economics, Project Management and Professional Ethics</td>
<td>PC-C</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>MCSExx</td>
<td>Elective -VII</td>
<td>PC-E</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>3</td>
<td>MCSE32</td>
<td>Internship/Industrial Training</td>
<td>IN</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>MCSE33</td>
<td>Project Work-I</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Internship:** The department should prepare Gantt chart with milestones, deliverables, evaluation and maintain weekly dairy signed by both Internal and External Guide.

**Project Work-I:** Seminar on: problem definition, literature survey and methodology to be used.

### IV SEMESTER

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Name</th>
<th>Category</th>
<th>Credits</th>
<th>Contact Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>MCS41</td>
<td>Project Work-II</td>
<td>PC-C</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Electives

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stream 1</strong></td>
<td></td>
</tr>
<tr>
<td>MCSE01</td>
<td>Advances in Computer Networks</td>
</tr>
<tr>
<td>MCSE02</td>
<td>Wireless Ad hoc Networks</td>
</tr>
<tr>
<td>MCSE03</td>
<td>Distributed Systems</td>
</tr>
<tr>
<td>MCSE04</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>MCSE05</td>
<td>Software Defined Networks</td>
</tr>
<tr>
<td>MCSE06</td>
<td>Advances in Storage Area Networks</td>
</tr>
<tr>
<td>MCSE07</td>
<td>Protocol Engineering</td>
</tr>
<tr>
<td>MCSE08</td>
<td>Network Management</td>
</tr>
<tr>
<td><strong>Stream 2</strong></td>
<td></td>
</tr>
<tr>
<td>MCSE09</td>
<td>Multi core Architecture and Programming</td>
</tr>
<tr>
<td>MCSE10</td>
<td>Embedded Systems</td>
</tr>
<tr>
<td>MCSE11</td>
<td>Compiler Optimization Techniques</td>
</tr>
<tr>
<td>MCSE12</td>
<td>Advanced Software Engineering</td>
</tr>
<tr>
<td>MCSE13</td>
<td>Software Oriented Architecture</td>
</tr>
<tr>
<td>MCSE14</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>MCSE15</td>
<td>Web Technologies</td>
</tr>
<tr>
<td>MCSE16</td>
<td>Advances in Digital Image Processing</td>
</tr>
<tr>
<td>MCSE17</td>
<td>Advances in Operating Systems</td>
</tr>
<tr>
<td><strong>Stream 3</strong></td>
<td></td>
</tr>
<tr>
<td>MCSE18</td>
<td>Cryptography and Security</td>
</tr>
<tr>
<td>MCSE19</td>
<td>Information and network security</td>
</tr>
<tr>
<td>MCSE20</td>
<td>Digital Forensics and cyber crime</td>
</tr>
<tr>
<td>MCSE21</td>
<td>Applied Cryptography</td>
</tr>
<tr>
<td>MCSE22</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>MCSE23</td>
<td>Software Testing</td>
</tr>
<tr>
<td>MCSE24</td>
<td>Privacy and security in Online Social Media</td>
</tr>
<tr>
<td>MCSE25</td>
<td>Start-up Engineering</td>
</tr>
<tr>
<td><strong>Stream 4</strong></td>
<td></td>
</tr>
<tr>
<td>MCSE26</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>MCSE27</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MCSE28</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>MCSE29</td>
<td>Social Network Analysis</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>MCSE30</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>MCSE31</td>
<td>Soft Computing</td>
</tr>
<tr>
<td>MCSE32</td>
<td>Computer System Performance Analysis</td>
</tr>
<tr>
<td>MCSE33</td>
<td>Future Skills 2020</td>
</tr>
<tr>
<td>MCSE34</td>
<td>Cyber Physical Systems</td>
</tr>
<tr>
<td>MCSE35</td>
<td>High Performance Computing</td>
</tr>
</tbody>
</table>

**Note:**

The total number of credits for all the elective courses is 4. The Faculty coordinator can choose to conduct a 1 credit integrated lab or 1 credit Tutorial for the course offered. The lab exercises and tutorial exercises will be formulated during teaching.
Advanced Engineering Mathematics

Course Code: MCS11          Credits: 4:0:0
Prerequisites: Engineering Mathematics I-IV          Contact Hours: 56
Course Coordinator/s: Dr. Govindaraju M V

Unit I

Unit II
Linear Transformations: Introduction to Linear transformations, Composition of matrix transformations, Rotation about the origin, Dilation, Contraction and Reflection, Kernel and Range, Change of basis.

Unit III

Unit IV

Unit V
Introduction to Queuing and Applications: Single server with infinite system capacity queuing models. M/M/1: \( \infty / FIFO \), K/FIFO, M/M/s: \( \infty / FIFO \), K/FIFO, M/G/1 Queuing system characteristics, Case studies.

Text Books:

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Solve the problems of algebraic, transcendental and ordinary differential equations using numerical methods. (PO1,3,4)
2. Find the Kernel and Range of Linear transformations. (PO1,3)
3. Express the probability distribution arising in the study of engineering problems and their applications. (PO1,3,4)
4. Apply the Markov Chain in prediction of future events. (PO1,3)
5. Apply and calculate the various parameters of the queuing models. (PO1,3)
Advances in Database Management Systems
Course Code: MCS12 Credits: 3:1:0
Prerequisites :DBMS Contact Hours:42
Course Coordinator/s: Dr. Seema S

Unit I

Unit II
Indexing and Hashing: Basic Concepts, Ordered Indices, B+-Tree Index Files, B+-Tree Extensions, Multiple-Key access, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Index Definition in SQL.

Unit III

Unit IV

Unit V
Distributed Databases: Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control in Distributed Databases, Availability, Distributed Query Processing, Heterogeneous Distributed Databases, Cloud-Based Databases, Directory Systems

Text Books:

Reference book:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Understand different database model and the overview of relation database model. (PO1,3,4)
2. Understand the methods of storing, managing and interrogating complex data and the background processes involved in query processing. (PO1,3,4)
3. Analyze background methods to optimizing the query. (PO1,3,4)
4. Understand the concept of distributed databases and parallel databases. (PO1,3,4)
5. Analyze the processing the queries in distributed and parallel databases. (PO1,3,4)
Software Development for Portable Devices Laboratory

Course Code: MCSL13  Credits: 0:0:1
Prerequisites: Nil  Contact Hours: 28
Course Coordinator/s: Mr. Pramod Sunagar

Experiments that are to be conducted as a part of the course:

1. Introducing different Android development tools and developing Hello World application.
2. Develop an android application to investigate the activity life cycle.
3. Develop an android application to investigate the fragments.
4. Develop an android application to create user interfaces with different layouts and views.
5. Develop an android application to create a Registration form using appropriate widgets.
6. Develop an android application to embed PickerViews in an activity.
7. Develop an android application on using implicit & explicit Intents.
8. Develop an android application to utilize Action bar.
9. Develop an android application to utilize Toasts and Notifications.
10. Develop an android application to work SQLite data storage and create a table.
11. Develop an android application to work SQLite data storage and perform various operations on the table.
12. Develop an android application to introduce content providers.
13. Developing applications to work with messaging and telephony services.
14. Develop an android application for creating location based service.

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Identify the various aspects of android platform and different android developer tools. (PO1,3,4)
2. Recognize the activity life cycle and fragment life cycle. (PO1,3,4)
3. Demonstrate the usage of different Views and ViewGroups. (PO1,3,4)
4. Demonstrate the knowledge of location-based services and notifications in applications. (PO1,3,4)
5. Develop the Android’s communication abilities using SMS and Telephony services. (PO1,3,4)
Python Programming Laboratory

Course Code: MCSL14  
Credits: 0:0:1  
Prerequisites: Nil  
Contact Hours: 28

Course Coordinator/s: Dr. Rajarajeswari S

Course Contents:

1. Python Basics
2. Control Structures
3. Functions
4. Strings, lists, list comprehensions
5. Tuples and dictionaries
6. Modules and packages
7. Object Oriented Concepts
8. Regular Expression
9. Programs on File I/O
10. Exceptions
11. Network Programming
12. GUI Programming
13. Design a simple game application using pygame
14. Game application Demo.

Text Books:
2. Problem Solving and Python Programming, E Balagurusamy

Reference Books:
1. Introduction to computer science using Python : A computational Problem solving focus, Charles dierbagh.
Course Outcomes (COs)

After the course, students should be able to:

1. Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms) (PO1,3,4)
2. Adequately use standard programming constructs: repetition, selection, functions, Composition, modules, aggregated data (arrays, lists, etc.) (PO1,3,4)
3. Understand and use object based software concepts (constructing OO software will be dealt with in the course Software Engineering) (PO1,3,4)
4. Create, debug and test a software application using the Python programming language (PO1,3,4)
5. Use library software for (e.g.) building a graphical user interface, web application, or Mathematical software (PO1,3,4)
Technical Seminar-I/II
Course Code: MCS15/MCS25
Prerequisites: Nil
Credits: 0:0:2

Rubrics for assessment of Seminar

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Distinguished (5)</th>
<th>Good (4)</th>
<th>Basic (3)</th>
<th>Unacceptable (1)</th>
<th>Score</th>
</tr>
</thead>
</table>
| Organization | - Extremely well organized.  
- Introduces the purpose of the presentation clearly and creatively.  
- Effectively includes smooth, clever transitions which are succinct but not broken up in order to connect key points  
- Student presents information in logical, interesting sequence which audience can follow.  
- Ends with an accurate conclusion showing thoughtful, strong organization. | - Generally well organized.  
- Introduces the purpose of the presentation clearly.  
- Includes transitions to connect key points and better transitions from idea to idea are noted.  
- Most information presented is in logical sequence; A few minor points may be confusing  
- Ends with a summary of main points showing some evaluation of | - Somewhat organized.  
- Introduces the purpose of the presentation  
- Includes some transitions to connect key points but there is difficulty in following presentation.  
- Student jumps around topics.  
Several points are confusing.  
- Ends with a summary or conclusion; little evidence of evaluating content | - Poorly organized  
- Does not clearly introduce the purpose of the presentation  
- Uses ineffective transitions that rarely connect points; cannot understand presentation because there is no sequence of information.  
- Presentation is broken and disjointed; no apparent logical order of presentation-Ends without a summary |
<p>| <strong>Content: Depth and Accuracy</strong> | <strong>-Speaker provides an accurate and complete explanation of key concepts and theories, drawing upon relevant literature.</strong>&lt;br&gt;Applications of theory are included to illuminate issues.&lt;br&gt;-Provides evidence of extensive and valid research on the selected topic, with multiple and varied sources.&lt;br&gt;-Combines and evaluates existing ideas to form new insights.&lt;br&gt;-Information completely accurate; all names and facts were precise and | <strong>-For the most part, explanations of concepts and theories are accurate and complete. Some helpful applications of theory are included.&lt;br&gt;-Presents evidence of valid research on the selected topic, with multiple sources.&lt;br&gt;-Combines existing ideas to form new insights.&lt;br&gt;-No significant errors are made; a few inconsistencies or errors in information.&lt;br&gt;-Level of presentation is</strong> | <strong>-Explanations of concepts and/or theories are inaccurate or incomplete. Little attempt is made to tie in theory. There is a great deal of information that is not connected to the current presentation.&lt;br&gt;-Presents evidence of research on the selected topic, with sources.&lt;br&gt;-Combines existing ideas.&lt;br&gt;-Few errors are made to distract a knowledgeable listener, but some information is accurate.&lt;br&gt;-Level of presentation is</strong> | <strong>-No reference is made to literature or theory. Presentation is not clear; information that does not support presentation in any way is unnecessarily included.&lt;br&gt;-Presents little or no evidence of valid research on the selected topic.&lt;br&gt;-Shows little evidence of the combination of ideas.&lt;br&gt;-Information included is sufficiently inaccurate that indicates absence of accurate</strong> |</p>
<table>
<thead>
<tr>
<th>Creativity</th>
<th>Use of Communication Aids</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td></td>
</tr>
<tr>
<td>- Level of presentation is appropriate for the audience.</td>
<td>generally appropriate.</td>
</tr>
<tr>
<td>- Portions of presentation are too elementary or too sophisticated for audience.</td>
<td>information.</td>
</tr>
<tr>
<td>- Presentation consistently is too elementary or too sophisticated for the audience.</td>
<td></td>
</tr>
<tr>
<td>- Uses the unexpected to full advantage; very original, clever, and creative approach that captures audience's attention.</td>
<td>- Some originality apparent; clever at times; good variety and blending of materials/media.</td>
</tr>
<tr>
<td>- Little or no variation; a few original touches but for the most part material presented with little originality or interpretation.</td>
<td>- Bland and predictable. Repetitive with little or no variety; little creative energy used.</td>
</tr>
<tr>
<td>- Occasional use of graphics that rarely support presentation; visual aids were not useful or clear, time wasting use of multimedia.</td>
<td>- Student uses superfluous graphics, no graphics, or graphics that are so poorly prepared that they detract from the presentation.</td>
</tr>
<tr>
<td>- Graphics designed reinforce presentation and maximize audience understanding; use of media is varied and appropriate with media not being added simply for the sake of use.</td>
<td>- While graphics relate and aid presentation, media are not as varied and not as well connected to the presentation.</td>
</tr>
<tr>
<td>Use of Language</td>
<td>-Poised, clear articulation; proper volume; steady rate; enthusiasm; confidence; speaker is clearly comfortable in front of the group. -Presentation has no misspellings or grammatical errors.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Eye Contact</td>
<td>-Maintains eye contact; seldom returning to notes; presentation is like a planned conversation.</td>
</tr>
<tr>
<td>Viva Voce</td>
<td>-Demonstrates extensive knowledge of the topic by responding confidently, precisely and appropriately to all</td>
</tr>
<tr>
<td>Audience questions</td>
<td>Document is fully compliant with required rules and structure. Document uses highly appropriate language and style.</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Report</td>
<td></td>
</tr>
<tr>
<td>Regularity</td>
<td>Reports to guide regularly for seminar discussion</td>
</tr>
<tr>
<td>Overall Presentation</td>
<td>Excellent</td>
</tr>
</tbody>
</table>

24
Data Analytics Using R

Course Code: MCS21 Credits: 4:0:0
Prerequisites: Nil Contact Hours: 56
Course Coordinator: Dr. S Rajarajeswari

Unit I
Introduction to big Data: Big Data Overview with typical examples, What is Data Science, Introduction to Big Data Analytics, Data Analytics Life cycle, Domain-specific Life cycle.

Unit II
Data, Data Range, Data Management, Data Mining, Data Munging, Wrangling and Cleaning, Databases and Relational Algebra, Parallel Data Bases, Parallel Query Processing, Mean and Standard Deviation, Estimation-Only Analyses, Use Case: Watching Data Trends with Google Ngrams, Use Case: Estimating Move Preferences.

Unit III
Introduction to Analytics - introduction to machine learning, Supervised learning overview, simple nearest neighbor, decision trees/forests, regression, Unsupervised learning: k-means, multi-dimensional scaling Graph Analytics: PageRank, community detection, recursive queries, iterative processing, Text Analytics: latent semantic analysis, Visualization & visual data analytics.

Unit IV

Unit V
Case Studies and Illustrations: IBM BigInsights, BigSheets, and Netezza Customer Intelligence, RainStor Big Data Analytics on Hadoop - The Industry's First Enterprise-Class Database Running Natively on Hadoop, DataStax (Including Coverage of the Free "Community Edition" of DataStax's Cassandra Implementation, with OpsCenter), Microsoft's Big Data Solution, Jigsaw: Visualization for Investigative Analysis and latest state of the art examples.
Text Books:
2. Lin, J., & Dyer, C. “Data-intensive text processing with MapReduce”, 2010, (Chapters: 2 and 5)

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. To take typical Big Data, filter and wrangle the data. (PO1,3,4)
2. To recognize the domain for analysis. (PO1,3,4)
3. To identify the type of domain-specific analytics to be deployed. (PO1,3,4)
4. To carry out mathematical approach in analytics. (PO1,3,4)
5. To be ready as a Data Scientist leading towards suitable job in analytics in industry or to pursue research towards higher degree. (PO1,3,4)
Advanced Algorithms

Course Code: MCS22
Credits: 3:0:1
Prerequisites: Knowledge of Analysis and Design of Algorithm
Contact Hours: 42+28

Course Coordinator: Dr. Jagadish S Kallimani

Unit I


Unit II


Unit III


Unit IV


Unit V


Text Books:

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Devise recurrence relations and amortized cost of various operations. (PO1,3,4)
2. Illustrate graph algorithms such as Bellman-Ford, Shortest path, bipartite matching, B-trees, Red-Black trees and hashing techniques. (PO1,3,4)
3. Identify the methods for solving modular linear equations, Chinese remainder theorem and RSA cryptosystem, types of heaps such as Binomial and Fibonacci heaps. (PO1,3,4)
4. Assess the string matching algorithms such as Boyer-Moore and Knuth-Morris-Pratt algorithm. (PO1,3,4)
5. Compose mathematical models, objective functions and constraints to solve algorithmic puzzles. (PO1,3,4,5)
IOT and Cloud Computing Laboratory
Course Code: MCSL23 Credits: 0:0:1
Prerequisites: Nil Contact Hours: 28
Course Coordinator/s: Mrs. Aparna R

Internet of Things Laboratory:
Internet of Things Laboratory provides one of the simplest ways to design and implement various IoT applications. IoT is a flexible platform for experimental research that uses connected devices in homes and beyond.
- Interconnection of devices and implementation of application scenarios.
- Deployment and monitoring of field studies and analysis of data from experiments.
- Sharing of data, code to evaluate ideas in a diverse set.
Projects can be designed in various IoTs domain like
1. Home Automation- Smart Lighting, Smart Appliances, Intrusion Detection, Smoke/Gas Detectors.
2. Cities - Smart Parking, Smart Lighting, Smart Roads, Structural Health Monitoring, Surveillance Emergency Response.
5. Retail - Inventory Management, Smart Payments, Smart Vending Machines.
7. Agriculture - Smart Irrigation, Green House Control.
The above mentioned projects can be developed using microcontrollers like Arduino Boards and Raspberry Pi.
Course Delivery: This course is delivered through presentations and miniproject.

Cloud Computing Laboratory
List of Experiments:
Use Eucalyptus or Open Nebula or equivalent to set up the cloud and demonstrate:
1. Find procedure to run the virtual machine of different configuration. Check how many virtual machines can be utilized at particular time.  
2. Find procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine.  
3. Install a C compiler in the virtual machine and execute a sample program.  
4. Show the virtual machine migration based on the certain condition from one node to the other.  
5. Find procedure to install storage controller and interact with it.  
6. Find procedure to set up the one node Hadoop cluster.  
7. Mount the one node Hadoop cluster using FUSE.  
8. Write a program to use the API’s of Hadoop to interact with it.  
9. Write a wordcount program to demonstrate the use of Map and Reduce tasks.

**List of Experiments:**

1. **SaaS: Google Drive**  
   - With your Gmail account, create a spreadsheet to share with the people at the same table, invite them.  
   - See how you can simultaneously edit the document you just created

2. **PaaS: google Maps**  
   http://maps.google.com/maps/api/staticmap?center=Eiffel+Tower&zoom=12 &size=512x512&sensor=false

3. **Design Virtual Machine using VM player and test Client server application using Virtual Machine**

4. **Design Virtual Machine using VM player and test Client server application using Virtual Box**

5. **Paas – Deploy Applications to google App Engine - simple web applications**

6. **Paas – Deploy Applications to google App Engine - web applications with database**

7. **Deploy Applications to cloud foundry using VMC**

8. **Deploy Applications to cloud foundry using Micro cloud foundry**

9. **Deploy Applications to cloud foundry using Eclipse**

10. **To Set up a Hadoop Cluster – Single Node**

11. **To Set up a Hadoop Cluster – Multi Node**

12. **Execute Map Reduce Programs in Hadoop Cluster**

13. **Study of Future Grid**
Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Design applications that will communicate with IoT hardware and software. (PO1,3,4)
2. Explore specific IoT domain like Network, communications, Management infrastructure, Services applications development and Human interaction. (PO1,3,4)
3. Examine an IoT offering of the project in terms of IoT levels and Protocols. (PO1,3,4)
4. Describe different kinds of Internet-connected product concepts. (PO1,3,4)
Data Analytics Lab

Course Code: MCS L24  
Credits: 0:0:1  
Prerequisites: Nil  
Contact Hours: 28  
Course Coordinator/s: Mr. Srinidhi H

1. HDFS  
Start by reviewing HDFS. You will find that its composition is similar to your local Linux file system. You will use the Hadoop fs command when interacting with HDFS.  
- Review the commands available for the Hadoop Distributed File System  
- Copy file foo.txt from local disk to the user’s directory in HDFS  
- Get a directory listing of the user’s home directory in HDFS  
- Get a directory listing of the HDFS root directory  
- Display the contents of the HDFS file user/fred/bar.txt  
- Move that file to the local disk, named as baz.txt  
- Create a directory called input under the user’s home directory  
- Delete the directory input old and all its contents  
- Verify the copy by listing the directory contents in HDFS

2. MapReduce  
- Create a JOB and submit to cluster  
- Track the job information  
- Terminate the job  
- Counters in MR Jobs with example  
- Map only Jobs and generic map examples  
- Distributed cache example  
- Combiners, Secondary sorting and Job chain examples

3. MapReduce (Programs) Using movie lens data  
- List all the movies and the number of ratings  
- List all the users and the number of ratings they have done for a movie  
- List all the Movie IDs which have been rated (Movie Id with at least one user rating it)  
- List all the Users who have rated the movies (Users who have rated at least one movie)  
- List of all the User with the max, min, average ratings they have given against any movie
• List all the Movies with the max, min, average ratings given by any user

4. Extract facts using Hive. Hive allows for the manipulation of data in HDFS using a variant of SQL. This makes it excellent for transforming and consolidating data for load into a relational database. In this exercise you will use HiveQL to filter and aggregate click data to build facts about user’s movie preferences. The query results will be saved in a staging table used to populate the Oracle Database. The moveapp_log_json table contains an activity column. Activity states are as follows:

• RATE_MOVIE
• COMPLETED_MOVIE
• PAUSE_MOVIE
• START_MOVIE
• BROWSE_MOVIE
• LIST_MOVIE
• SEARCH_MOVIE
• LOGIN
• LOGOUT
• INCOMPLETE_MOVIE

hive> SELECT * FROM movieapp_log_json LIMIT 5;
hive> drop table movieapp_log_json;
hive> CREATE EXTERNAL TABLE movieapp_log_json ( custId INT, movieId INT, genreId INT, time STRING, recommended STRING, activity INT, rating INT, price FLOAT ) ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.JsonSerde' LOCATION '/user/oracle/moviework/applog/';
hive> SELECT * FROM movieapp_log_json LIMIT 20;
hive> SELECT MIN(time), MAX(time) FROM movieapp_log_json

1. PURCHASE_MOVIE Hive maps queries into Map Reduce jobs, simplifying the process of querying large datasets in HDFS. HiveQL statements can be mapped to phases of the Map Reduce framework. As illustrated in the following figure, selection and transformation operations occur in map tasks, while aggregation is handled by reducers. Join operations are flexible: they can be performed in the reducer or mappers depending on the size of the leftmost table. 1. Write a query to select only those clicks which correspond to starting, browsing, completing, or purchasing movies. Use a CASE statement to transform the RECOMMENDED column into integers where ‘Y’ is 1 and ‘N’ is 0. Also, ensure GENREID is not null. Only include the first 25 rows.
2. Write a query to select the customer ID, movie ID, recommended state and most recent rating for each movie.
3. Load the results of the previous two queries into a staging table. First, create the staging table:
4. Next, load the results of the queries into the staging table.
5. Extract sessions using Pig While the SQL semantics of HiveQL are useful for aggregation and projection, some analysis is better described as the flow of data through a series of sequential operations. For these situations, Pig Latin provides a convenient way of implementing data flows over data stored in HDFS. Pig Latin statements are translated into a sequence of Map Reduce jobs on the execution of any STORE or DUMP command. Job construction is optimized to exploit as much parallelism as possible, and much like Hive, temporary storage is used to hold intermediate results. As with Hive, aggregation occurs largely in the reduce tasks. Map tasks handle Pig’s FOREACH and LOAD, and GENERATE statements. The EXPLAIN command will show the execution plan for any Pig Latin script. As of Pig 0.10, the ILLUSTRATE command will provide sample results for each stage of the execution plan. In this exercise you will learn basic Pig Latin semantics and about the fundamental types in Pig Latin, Data Bags and Tuples.
   1. Start the Grunt shell and execute the following statements to set up a dataflow with the click stream data. Note: Pig Latin statements are assembled into Map Reduce jobs which are launched at execution of a DUMP or STORE statement.
   2. Group the log sample by movie and dump the resulting bag.
      1. Add a GROUP BY statement to the sessionize.pig script to process the click stream data into user sessions.

**Reference Books:**
**Course Outcomes (COs):**
At the end of the course, students should be able to:
1. Describe big data and use cases from selected business domains. (PO1,3,4)
2. Explain NoSQL big data management. (PO1,3,4)
3. Install, configure, and run Hadoop and HDFS. (PO1,3,4)
4. Perform map-reduce analytics using Hadoop. (PO1,3,4)
5. Use Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data Analytics. (PO1,3,4)
Engineering Economics, Project Management and Professional Ethics
Course Code: MCS31  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Prof. Nagabhushana A M

Unit I

Unit II
Organizing and staffing the project office and the team: The Staffing Environment, Selecting the Project Manager: An Executive Decision, Skill Requirements for Project and Program Managers, Next Generation Project Managers, Duties and Job Descriptions, Selecting the Project Management Implementation Team, Management Functions: Project Authority, Team Building as an Ongoing Process, Dysfunctions of a Team, Employee–Manager Problems, Management Pitfalls, Conflicts: The Conflict Environment, Conflict Resolution, The Management of Conflicts, Conflict Resolution Modes, the variables for success: Predicting Project Success, Project Management Effectiveness.

Unit III

Unit IV

**Unit V**


**Reference Book:**

**Course Outcomes (COs):**
At the end of the course, the students will be able to:
1. Recognize issues in a realistic project scenario. (PO 1, 3, 4)
2. Employ work breakdown structures (WBS) in a project application. (PO 1, 3, 4)
3. Demonstrate the use of appropriate network scheduling techniques. (PO 1, 3, 4)
4. Produce a project proposal. (PO 1, 3, 4)
5. Discuss the implementation of a proposed plan. (PO 1, 3, 4)
ELECTIVES
 Advances in Computer Networks

Course Code: MCSE01 Credits: 4:0:0
Prerequisites: Computer Networks Contact Hours: 56
Course Coordinator/s: Dr. Monica R Mundada

Unit I

Unit II
Internetworking I: Switching and Bridging, Datagram’s, Virtual Circuit Switching, Source Routing, Bridges and LAN Switches, Basic Internetworking (IP), What is an Internetwork?, Service Model, Global Addresses, Datagram Forwarding in IP, sub netting and classless addressing, Address Translation (ARP), Host Configuration (DHCP), Error Reporting (ICMP), Virtual Networks and Tunnels.

Unit III
Internetworking- II: Network as a Graph, Distance Vector (RIP), Link State (OSPF), Metrics, The Global Internet, Routing Areas, Routing among Autonomous systems (BGP), IP Version 6 (IPv6), Mobility and Mobile IP

Unit IV

Unit V
Congestion Control and Resource Allocation Congestion-Avoidance Mechanisms, DEC bit, Random Early Detection (RED), Source-Based Congestion Avoidance. The Domain Name System (DNS), Electronic Mail (SMTP,POP,IMAP,MIME), World Wide Web (HTTP), Network Management (SNMP)
Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Explain basic computer network technology. (PO1,3,4)
2. Explain Data Communications System and its components. (PO1,3,4)
3. Identify the different types of network topologies and protocols. (PO1,3,4)
4. Identify the different types of network devices and their functions within a network. (PO1,3,4)
5. Illustrate the skills of subnet and routing mechanisms. (PO1,3,4)
Wireless Ad-Hoc Networks

Course Code: MCSE02  Credits: 4:0:0
Prerequisites: Computer Networks  Contact Hours: 56
Course Coordinator/s: Dr. Annapurna P Patil

Unit I

Unit II

Unit III

Unit IV

Unit V

Text Book:

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Understand of basic WSN technology and supporting protocols and technology. (PO1,3,4)
2. Identify medium access control protocols and address physical layer issues. (PO1,3,4)
3. Describe the knowledge routing protocols for sensor networks and main design issues. (PO1,3,4)
4. Describe knowledge of transport layer protocols for sensor networks, and design requirements. (PO1,3,4)
5. Describe Sensor management, sensor network middleware, operating systems. (PO1,3,4)
Distributed Systems

Course Code: MCSE03  Credits: 4:0:0
Prerequisites: OS  Contact Hours: 56
Course Coordinator/s: Sini Anna Alex

Course Contents:

Unit I
Introduction: Definition, Relation to computer system components, Motivation, Relation to parallel multiprocessor/multicomputer systems, Message-passing systems versus shared memory systems, Primitives for distributed communication, Synchronous versus asynchronous executions, Design issues and challenges.
A model of distributed computations: A distributed program, A model of distributed executions, Models of communication networks, Global state of a distributed system, Cuts of a distributed computation, Past and future cones of an event, Models of process communications
Logical time: Introduction, A framework for a system of logical clocks, Scalar time, Vector time, Efficient implementations of vector clocks, Jard–Jourdan’s adaptive technique, Matrix time, Virtual time, Physical clock synchronization: NTP.

Unit II
Global state and snapshot recording algorithms: Introduction, System model and definitions, Snapshot algorithms for FIFO channels, Variations of the Chandy–Lamport algorithm, Snapshot algorithms for non-FIFO channels, Snapshots in a causal delivery system, Monitoring global state, Necessary and sufficient conditions for consistent global snapshots, Finding consistent global snapshots in a distributed computation.
Terminology and basic algorithms: Topology abstraction and overlays, Classifications and basic concepts, Complexity measures and metrics, Program structure, Elementary graph algorithms, Synchronizers, Maximal independent set (MIS), Connected dominating set, Compact routing tables, Leader election, Challenges in designing distributed graph algorithms, Object replication problems.

Unit III
Message ordering and group communication: Message ordering paradigms, Asynchronous execution with synchronous communication, Synchronous program order on an asynchronous system, Group communication, Causal order (CO), Total order, A nomenclature for
multicast, Propagation trees for multicast, Classification of application-level multicast algorithms, Semantics of fault-tolerant group communication, Distributed multicast algorithms at the network layer.

**Termination detection:** Introduction, System model of a distributed computation, Termination detection using distributed snapshots, Termination detection by weight throwing, A spanning-tree-based termination detection algorithm, Message-optimal termination detection, Termination detection in a very general distributed computing model, Termination detection in the atomic computation model, Termination detection in a faulty distributed system.

**Unit IV**


**Unit V**

**Global predicate detection:** Stable and unstable predicates, Modalities on predicates, Centralized algorithm for relational predicates, Conjunctive predicates, Distributed algorithms for conjunctive predicates, Further classification of predicates.

**Consensus and agreement algorithms:** Problem definition, Overview of results, Agreement in a failure-free system (synchronous or asynchronous), Agreement in (message-passing) synchronous systems with failures, Agreement in asynchronous message-passing systems with failures, Wait-free shared memory consensus in asynchronous systems.

**Peer-to-peer computing and overlay graphs:** Introduction, Data indexing and overlays, Unstructured overlays, Chord distributed hash table, Content addressable networks (CAN), Tapestry, Some other challenges in P2P system design, Tradeoffs between table storage and route lengths, Graph

Text Book:

Reference Books:

Course Outcomes (COs):

At the end of the course, the students will be able to:
1. Identify the design issues and Challenges in building distributed systems. (PO1,3,4)
2. Explore different ways of managing time (clock) and recording global state of distributed computation. (PO1,3,4)
3. Analyze basic distributed graph algorithms, synchronizers, and practical graph problems, P2P overlay problems.(PO1,3,4)
4. Discuss ways to achieve various message ordering schemes and approaches for detecting termination of a distributed computation. (PO1,3,4)
5. Identify different assertion based, and tree based distributed algorithms to implement Distributed Mutual Exclusion. (PO1,3,4)
Cloud Computing

Course Code: MCSE04  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator: Mrs. Rajarajeswari S

Unit I
Introduction: Network centric computing and network centric content, Peer-to-peer systems, Cloud Computing: an old idea whose time has come, Cloud Computing delivery models & Services, Ethical issues, Cloud vulnerabilities, Challenges, Cloud Infrastructure: Amazon, Google, Azure & online services, open source private clouds. Storage diversity and vendor lock-in, intercloud, Energy use & ecological impact of data centers, service level and compliance level agreement, Responsibility sharing, user experience, Software licensing.

Unit II

Unit III

Unit IV
Cloud Resource Management and Scheduling: Policies and mechanisms for resource management, Applications of control theory to task scheduling on a cloud, Stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds, Coordination of specialized autonomic performance managers, A utility-based model for cloud-based web services, Resource bundling, combinatorial auctions, fair queuing, Start time fair queuing, borrowed virtual time, Cloud scheduling subject to deadlines.
Scheduling mapreduce applications subject to deadlines, Resource management and application scaling.

Unit V

Storage systems: Evolution, Storage models, file systems, databases, DFS, General parallel File system, GFS, Hadoop, Locks & Chubby, TPS, NOSQL, BigTable, Mega store.


Text Books:

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. To analyze the transformation that led to the evolution of Cloud computing, it's vulnerabilities and its impact. (PO1,3,4)
2. Design different workflows according to requirements and apply map reduce programming model and real time applications. (PO1,3,4)
3. Make performance comparison of virtual machines and optimization of virtualization. (PO1,3,4)
4. Develop policies and mechanisms for cloud resources and design algorithms for scheduling. (PO1,3,4)
5. Assess cloud Storage systems and identify cloud security, privacy issues, threats, the risks involved, its impact and provide solution. (PO1,3,4)
Software Defined Networks

Course Code: MCSE05  Credits: 4:0:0
Prerequisites: Data communications and Computer networks  Contact Hours: 56

Course Coordinator: Sanjeetha R
Prerequisites: Data communications and Computer networks

Course Contents:

Unit I
Introduction - Traditional Switch Architecture.
Why SDN- Evolution of Switches and Control Planes, Cost, SDN Implications for Research and Innovation, Data Center Innovation, Data Center Needs.

Unit II
How SDN Works Contd. - SDN Controller core modules, SDN controller interfaces, Existing controller implementations, potential issues with the SDN Controller, SDN Applications, Alternate SDN Methods – SDN via APIs, Benefits and Limitations of SDN via APIs, SDN via hypervisor based overlay networks.

Unit III
The OpenFlow Specification Contd. Openflow 1.4 additions – Bundles, Eviction and vacancy events, enhanced support for multiple controller, optical port support, and flow table synchronization.
Alternative Definitions of SDN - Potential Drawbacks of Open SDN, SDN via APIs – Legacy APIs in Network Devices, NETCONF/YANG, BGP-LS/PCE-P, REST, Examples of SDN via APIs, Ranking SDN via APIs, SDN via Hypervisor-Based Overlays – Overlay Controller, Overlay Operation, Examples of SDN via Hypervisor-Based Overlays, Ranking SDN via Hypervisor-Based Overlays, SDN via Opening Up the Device, Network Functions Virtualization, Alternatives Overlap and Ranking.

flow table synchronization, Alternatives Overlap and Ranking.

Unit IV


SDN in the Data Center - Data Center Demands – Overcoming Current Network Limitations, MAC address explosion, Number of VLANs, Spanning tree, adding, moving and deleting resources, Failure recovery, multitenancy, Tunneling Technologies for the Data Center.

Intents-Based Applications, Tunneling Technologies for the Data Center.

Unit V

SDN in the Data Center - Path Technologies in the Data Center, Ethernet Fabrics in the Data Center, SDN Use Cases in the Data Center, Comparison Of Open SDN, Overlays, and APIs.

SDN in Other Environments - Wide Area Networks, Service Provider and Carrier Networks, Campus Networks, Mobile Networks, Optical Networks.

Network Functions Virtualization - Definition Of NFV, What Can We Virtualize? SDN Vs NFV, When Should NFV Be Used With SDN?, In-Line Network Functions, SDN Applied To Server Load-Balancing, Firewalls and Intrusion Detection.

SDN Applications - Application Types, a Simple Reactive Java Application - Blacklisting Hostnames and IP Addresses, Offloading Flows in the Data Center.

Optical Networks, Firewalls and Intrusion Detection, Offloading Flows in the Data Center.
Text Book:


Reference Book:


Course Outcomes(COs):

At the end of the course, students should be able to:

1. Describe the fundamental characteristics of SDN. (PO1,3,4)
2. Differentiate between various OpenFlow specifications. (PO1,3,4)
3. Identify different ways of implementing SDN. (PO1,3,4)
4. Compare and contrast different types of controller models in SDN. (PO1,3,4)
5. Illustrate use of SDN in Data Centers and other environments(PO1,3,4)
Advances in Storage Area Networks
Course Code: MCSE06 Credits: 4:0:0
Prerequisites: Computer Networks, Computer Organization, Operating Systems Contact Hours: 56
Course Coordinator/s: Dr. Divakar Harekal

Course Contents:

Unit I

Unit II

Unit III
Fiber Channel Storage Area Networks: FC Overview, Evolution, Components, FC Connectivity, Ports, FC Architecture, Fabric Services, Login Types, Zoning, FC Topologies, Virtualization in SAN. IP SAN and FCoE: iSCSI, FCIP, FCoE.

Unit IV

Unit V
Business Continuity: Information Availability, Terminology, Planning Lifecycle, Failure Analysis, Impact Analysis, Challenges, Adoption Considerations. Securing the Storage Infrastructure: Framework, Risk Triad, Domains Managing the Storage Infrastructure: Monitoring, Management...
Activities, Management Challenges, Information Lifecycle Management, Storage Tiering.

Text Book:


Reference Books:


Course Outcomes (COs):

At the end of the course, student should be able to:

1. Identify the need for storage centric network and its benefits of its adoption (PO1,3,4)
2. Design a storage solution for an application depending on the IOPS and RAID requirements (PO1,3,4)
3. Have an understanding of the Fiber channel stack and working of the different layers (PO1,3,4)
4. Have an understanding of NAS, object oriented storage and backup and recovery (PO1,3,4)
5. Have a business continuity plan and ILM of an enterprise (PO1,3,4)
Protocol Engineering

Course Code: MCSE07
Credits: 4:0:0
Prerequisites: Nil
Contact Hours: 56
Course Coordinator/s: Dr. Annapurna P Patil

Unit I


Unit II


Unit III


Unit IV

52

Unit V

Text Book:

Reference Book:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Recall the structure of TCP/IP protocol suite. (PO1,3,4)
2. Identify different phases of protocol engineering. (PO1,3,4)
3. Design SDL based protocol specifications for various protocols. (PO1,3,4)
4. Design and generate test sequences using different methods. (PO1,3,4)
5. Demonstrate different ways of protocol synthesis. (PO1,3,4)
Network Management

Course Code: MCSE08  Credits: 4:0:0
Prerequisites: Computer Networks  Contact Hours: 56
Course Coordinator/s: Dr. Monica R Mundada

Course Contents:

Unit I

Unit II

Unit III

Unit IV
SNMP Management: RMON: What is Remote Monitoring?, RMON SMI and MIB, RMON1, RMON2, ATM Remote Monitoring, a Case Study of Internet Traffic Using RMON, ATM Networks.

Unit V
Text Book:


Course Outcomes (COs):

At the end of the course, students should be able to:

1. Identify five major functional areas of network management. (PO-1,3,4)
2. Explore SNMP Management Information, Standard MIBs and SNMP protocol. (PO-1,3,4)
3. Defend the need for security in networking monitoring and control. (PO-1,3,4)
4. Devise Remote network monitoring System for statistics collection, alarms and filters. (PO-1,3,4)
5. Summarize complete network management plan for a moderate to large network enterprise through case studies. (PO-1,3,4)
Multi-core Architecture and Programming

Course Code: MCSE09  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Mr. Mallegowda M

Unit I


Unit II


Unit III


Unit IV

Openmp: A Portable Solution For Threading: Challenges In Threading A Loop, Loop-Carried Dependence, Data-Race Conditions, Managing Shared And Private Data, Loop Scheduling And Portioning, Effective Use Of Reductions, Minimizing Threading Overhead, Work-Sharing Sections, Performance-Oriented Programming, Using Barrier And No Wait,
Interleaving Single-Thread And Multi-Thread Execution, Data Copy-In And Copy-Out, Protecting Updates Of Shared Variables, Intel Task Queuing Extension To Openmp, Openmp Library Functions, Openmp Environment Variables, Compilation, Debugging, Performance.

Unit V

Text Books:

Reference Book:

Course Outcomes (COs):
At the end of the course, the students should be able to:
1. Identify performance related parameters in the field of Computer Architecture. (PO1,3,4)
2. Identify the limitations of ILP and the need for multi-core architectures. (PO1,3,4)
3. Solve the issues related to multiprocessing and suggest solutions. (PO1,3,4)
4. Point out the salient features of different multi-core architectures and how they exploit parallelism. (PO1,3,4)
5. Understand the concept of multi threading and OPENMP. (PO1,3,4)
Embedded Systems

Course Code: MCSE10
Credits: 4:0:0
Prerequisites: Nil
Contact Hours: 56
Course Coordinator/s: Dr. Divakar Harekal V

Unit I
Embedded Computing: Introduction, Complex Systems And Microprocessors

Unit II
Programming Model, Operation Modes And States, Architecture, Registers And Special Registers Behaviors Of The Application Program Status Register (APSR), Memory System Overview, Stack Memory, Operations Introduction To Cortex-M0 Programming, Instruction Set.

Unit III
Instruction Usage Examples, Implementation Of Various Structures Like Loop, Switch, Function, Subroutine, Memory System, Exceptions And Interrupts, Interrupt Control And System Control.

Unit IV

Unit V

Text Books:
Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Identify embedded system requirements and goals.(PO1,3,4)
2. Describe Cortex M0 advantages for designing embedded systems. (PO1,3,4)
3. Evaluate Cortex M0 in assembly instructions and write embedded C programs using CMSIS features. (PO1,3,4)
4. Understand the memory mapping of Cortex M0 architecture. (PO1,3,4)
5. Compare the working of various sensors and actuators and their interface with microcontrollers. (PO1,3,4)
Compiler Optimization Techniques

**Course Code:** MCSE11  
**Credits:** 4:0:0  
**Prerequisites:** Nil  
**Contact Hours:** 56  
**Course Coordinator/s:** Dr. Jagadish S Kallimani

**Unit I**
Introduction, Lexical analysis, Compilers; Analysis of Source Program; The Phases of a Compiler; Cousins of the Compiler; The grouping of phases; Compiler- Construction tools. Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.

**Unit II**
Syntax Analysis, The Role of the Parser; Context-free Grammars; Writing a Grammar; Top-down Parsing; Bottom-up Parsing. Syntax Analysis, Operator-Precedence Parsing; LR Parsers; Using ambiguous grammars; Parser Generators.

**Unit III**
Syntax-Directed Translation, Syntax-Directed definitions; Constructions of Syntax Trees; Bottom-up evaluation of S-attributed definitions; L-attributed definitions; Top-down translation, Run-Time Environments, Source Language Issues; Storage Organization; Storage-allocation strategies, Storage-allocation in C; Parameter passing.

**Unit IV**
Intermediate Code Generation, Intermediate Languages; Declarations; Assignment statements; Boolean Expressions; Case statements; Back patching; Procedure calls, Code Generation, Issues in the design of Code Generator; The Target Machine; Run-time Storage Management; Basic blocks and Flow graphs; Next-use information; A Simple Code Generator; Register allocation and assignment; The dag representation of basic blocks; Generating code from dags.

**Unit V**
Code Optimization, Compiler Development, Code Optimization: Introduction; The principal sources of optimization; Peephole optimization; Optimization of basic blocks; Loops in flow graphs, Compiler Development: Planning a compiler; Approaches to compiler development; The compiler development environment; Testing and maintenance.

**Reference Books:**
(Chapters 1, 3.1 to 3.4, 4, 5.1 to 5.5, 7, 8, 9.1 to 9.9, 10.1 to 10.5, 11)

**Course Outcomes (COs):**
At the end of the course, the students will be able to:
1. Construct lexical analyzer to recognize inputs using patterns. (PO1,3, 4)
2. Devise different types of syntax analyzers using grammars. (PO1,3, 4)
3. Illustrate syntax-directed translation schemes for grammars. (PO 1, 3, 4)
4. Formulate intermediate code generators for programming statements. (PO1, 3, 4)
5. Develop assembly language code for the given optimized intermediate codes. (PO1, 3, 4)
Advanced Software Engineering

Course Code: MCSE12  Credits: 4:0:0
Prerequisites: Basic Concepts of Software Engineering  Contact Hours: 56

Course Coordinator/s: Dr. Annapurna P Patil

Unit I

Agile development: What is agile? Agility and cost of change; What is an agile process? Extreme programming; Other agile process models. Design Concepts: Design process, Design Concepts, Design Models. Web Application Design: Web application design quality; Design quality and design pyramid; Interface design; Aesthetic design; Content design; Architecture design; Navigation design; Component-level design; Object-oriented hypermedia design method.

Unit II

Formal Modeling and verification: The cleanroom strategy; Functional specification; Cleanroom design; Cleanroom testing; Formal methods: Concepts; Applying mathematical notation for formal specification; Formal specification languages. Software Project Management: The management spectrum; The management of people, product, process and project; The W5HH Principle; Critical practices. Estimation for Software Projects: Software project estimation; Decomposition techniques, Examples; Empirical estimation models; Estimation for Object-Oriented projects; specialized estimation techniques; the make / buy decision.

Unit III

Software Project Scheduling: Basic concepts and principles of project scheduling; Defining task set and task network; Scheduling; Earned value analysis. Risk Management: Reactive versus proactive strategies; Software risks; risk identification; Risk projection; Risk refinement; Risk mitigation, monitoring and management; The RMMM plan.

Unit IV

Maintenance and Reengineering: Software maintenance; Software supportability; Reengineering; Business process reengineering; Software reengineering; Reverse engineering; Restructuring; Forward engineering; The economics of reengineering. Software Process Improvement (SPI): Approaches to SPI; Maturity models; The SPI process; The CMMI; The People CMM; Other SPI frameworks: SPICE, Bootstrap, PSP and TSP, ISO; SPI return on investment.

Unit V
Software Configuration Management (SCM): Basic concepts; SCM repository; The SCM process; Configuration management for web applications; SCM standards. Product Metrics: A framework for product metrics; Metrics for requirements model, design model, source code, testing and maintenance; Design metrics for web applications. Process and Project Metrics: Basic concepts; Software measurement; Metrics for software quality; Integrating metrics within the software process; Metrics for small organizations; Establishing a software metrics program.

Laboratory:
The Software Engineering Lab has been designed for students to be exposed to the following:

- To Understand and realize the concept of Software Engineering and UML using any case studies.
- The students would be exposed to experiment with the following concepts using any of the tools available:
  - Identifying the Requirements from Problem Statements
  - Modeling UML Use Case Diagrams
  - Identifying Domain Classes from the Problem Statements
  - State chart and Activity Modeling
  - Modeling UML Class Diagrams and Sequence diagrams
  - Estimation of Test Coverage Metrics and Structural Complexity
  - Designing Test Suites for the case study
- The Students would also be required to Prepare a Work Breakdown Structure (WBS) and a mini project plan with PERT and Gantt charts

Text Book:

Reference Books:
Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Apply the concept of agile development and design any web application using the principles learnt. (PO 1,3,4)
2. Identify the concepts of Formal Modeling and verification methods and also the process of Project management and estimation. (PO 1,3,4)
3. Demonstrate the activities involved in Software Scheduling, Risk management and reverse engineering and Maintenance task. (PO 1,3,4)
4. Identify the approaches to Software Process Improvement. (PO 1,3,4)
5. Understand the concepts of Software Configuration Management, Product and Process metrics. (PO 1,3,4)
Software Oriented Architecture

Course Code: MCSE13  
Credits: 4:0:0

Prerequisites: Nil  
Contact Hours: 56

Course Coordinator/s: Dr. Divakar Harekal V

UnitI


UnitII


UnitIII


UnitIV


UnitV


Reference Books:
**Course Outcomes (COs):**

At the end of the course, the students will be able to:

1. Design, develop and test Web services. (PO 1,3,4)

2. Learn standards related to Web services: Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Description, Discovery and Integration (UDDI). (PO 1,3,4)

3. Learn basic principles of Service-Oriented Architecture and apply these concepts to develop a sample application. (PO 1,3,4)

4. Conceptually model Web services and formulate specifications of them in the Resource Description Framework (RDF) and the Web Ontology Language (OWL). (PO 1,3,4)

5. Evaluate emerging and proposed standards for the main components of Web services architectures. (PO 1,3,4)
Information Retrieval

Course Code: MCSE14 Credits: 4:0:0
Prerequisites: Nil Contact Hours: 56
Course Coordinator/s: Ms. Vandana Sardar Sudhakar

Unit I


Unit II


Unit III


Unit IV

Text Operations: Introduction, Document preprocessing, Document clustering, Text compression, comparing text compression techniques. Indexing and Searching: Introduction; Inverted Files; Other indices for text; Boolean queries; Sequential searching; Pattern matching; Structural queries; Compression. Parallel and Distributed IR: Introduction, Parallel IR, Distributed IR.

Unit V

User Interfaces and Visualization: Introduction, Human-Computer interaction, the information access process, Starting pints, Query specification, Context, Using relevance judgments, Interface support for the search process. Searching the Web: Introduction, Challenges, Characterizing the web, Search engines, Browsing, Meta searchers, finding the needle in the haystack, searching using hyperlinks.

Text Book:

Reference Books:

Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Distinguish between data and information retrieval systems and explain different classical IR models. (PO 1,3,4)
2. Assess the performance of information retrieval systems and use different querying languages and protocols. (PO 1,3,4)
3. Perform query operations and recognize the use of metadata and markup languages. (PO 1,3,4)
4. Explain various text operations, indexing and search techniques, and the basics of parallel and distributed IR. (PO 1,3,4)
5. Discuss the concepts of user interfaces for IR applications and web search techniques. (PO 1,3,4)
Web Technologies

Course Code: MCSE15
Prerequisites: Nil
Course Coordinator/s: Mrs. J Geetha

Unit I

Unit II
HTML 5: Detecting HTML 5 features – Canvas, video, local storage, web workers, offline applications, geo-location, placeholders, and input types. What does it all mean – doc type, root, headers, articles, dates and times, navigation and footers. Let’s call it a drawing surface – Simple shapes, canvas, Paths, texts, gradients and images. The past, present and future of local storage for web applications, A Form of madness – place holders, autofocus fields, email, web addresses, numbers as spin boxes and sliders, date and color pickers, search boxes.

Unit III

Unit IV
AJAX-II: Syndication with RSS and Atom – RSS, Atom, XParser, Creating a news ticker, Web search with RSS. JSON – Array, object, mixing literals, syntax, encoding/decoding, JSON versus XML, server-side JSON tools. COMET: HTTP streaming – request delays, file modification example, using Iframes, browserspecific approaches, server-sent DOM events, connection management and server-side support.

Unit V
Mashups and Web services: The rise of mashups, geocoding, Google maps API. Introduction to Service Oriented Architecture, Combining protocols to build Web services – clarifying web services, REST Services, WS-* Web services using SOAP and WSDL, REST vs WS-* services.
Reference Books:

Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Develop a dynamic webpage by the use of java script and DHTML. (PO 1,3,4)
2. Write a well formed / valid XML document. (PO 1,3,4)
3. Connect a java program to a DBMS and perform insert, update and delete operations on DBMS table. (PO 1,3,4)
4. Write a server side java application called Servlet to catch form data sent from client, process it and store it on database. (PO 1,3,4)
5. Write a server side java application called JSP to catch form data sent from client and store it on database. (PO 1,3,4)
Advances in Digital Image Processing

Course Code: MCSE16  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Mrs. Veena G S

Unit I

Unit II

Unit III

Unit IV

Unit V

Reference Books:
**Course Outcomes (COs):**
At the end of the course, students should be able to:
1. Examine various types of images, intensity transformations and applying various filtering techniques. (PO1,3,4)
2. Show how higher-level image concepts such as edge detection, segmentation representation can be implemented and used. (PO1,3,4)
3. To manipulate both binary and grayscale digital images using morphological filters and operators to achieve a desired result. (PO1,3,4)
4. Apply image processing algorithms in practical applications. (PO1,3,4)
Advances in Operating Systems

Course Code: MCSE17  Credits: 4:0:0
Prerequisites: Operating Systems  Contact Hours: 56
Course Coordinator/s: Dr. T N R Kumar

Unit I


Unit II


Unit III


Unit IV

Distributed Scheduling - Issues In Load Distributing, Components Of Load Distributing Algorithm - Stability, Algorithms - Performance Comparison, Selecting A Suitable Load Sharing Algorithm, Requirements For Load Distributing -Task Migration and Issues. Failure Recovery: Classification, Backward And Forward Error Recovery, Recovery In Concurrent Systems - Consistent Set Of Check Points - Synchronous And Asynchronous Check Pointing And Recovery, Check Pointing For Distributed Database Systems-Recovery In Replicated Distributed Databases.

Unit V

Laboratory Work:
(The following programs can be executed on any available and suitable platform)

1. Design, develop and execute a program using any thread library to create the number of threads specified by the user; each thread independently generates a random integer as an upper limit, and then computes and prints the number of primes less than or equal to that upper limit along with that upper limit.

2. Rewrite above program such that the processes instead of threads are created and the number of child processes created is fixed as two. The program should make use of kernel timer to measure and print the real time, processor time, user space time and kernel space time for each process.

3. Design, develop and implement a process with a producer thread and a consumer thread which make use of a bounded buffer (size can be prefixed at a suitable value) for communication. Use any suitable synchronization construct.

4. Design, develop, and execute a program to solve a system of n linear equations using Successive Over-relaxation method and n processes which use Shared Memory API.

5. Design, develop, and execute a program to demonstrate the use of RPC.

Reference Books:

Course Outcomes (COs):
At the end of the course, the students will be able to:

1. Implement a concurrent programming application using semaphores & monitors for process control. (PO 1,3,4)
2. Explain the basic concepts of Distributed Operating Systems and its architecture. (PO 1,3,4)
3. Implement deadlock avoidance, prevention & recovery. (PO 1,3,4)
4. Identify the Distributed resource management and design issues. (PO 1,3,4)
5. Implement various CPU scheduling, IPC memory management, recovery and concurrent algorithms. (PO 1,3,4)
Cryptography and Security

Course Code: MCSE18  Credits: 4:0:0
Prerequisites:  Contact Hours: 56
Course Coordinator/s: Sowmya B J

Unit I

Unit II
Confidentiality using conventional encryption - traffic confidentiality - key distribution - random number generation - Introduction to graph - ring and field - prime and relative prime numbers - modular arithmetic - Fermat’s and Euler’s theorem - primality testing - Euclid’s Algorithm - Chinese Remainder theorem - discrete algorithms.

Unit III

Unit IV
Unit V

Text Books:


Reference Books:


Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Provide security of the data over the network. (PO 1,3,4)
2. Do research in the emerging areas of cryptography and network security. (PO 1,3,4)
3. Implement various networking protocols. (PO 1,3,4)
4. Protect any network from the threats in the world. (PO 1,3,4)
5. Understand the IP security issues (PO 1,3,4)
Information and Network Security

Course Code: MCSE19  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Mrs. Meeradevi A Kawalgi

Unit I
Classical Encryption Techniques: Symmetric Cipher Model, Cryptography, Cryptanalysis and Brute-Force Attack, Substitution Techniques, Caesar Cipher, Mono alphabetic Cipher, Playfair Cipher, Hill Cipher, Polyalphabetic Cipher, One Time Pad. Block Ciphers and the data encryption standard: Traditional block Cipher structure, stream Ciphers and block Ciphers, Motivation for the feistel Cipher structure, the feistel Cipher, The data encryption standard, DES encryption, DES decryption, A DES example, results, the avalanche effect, the strength of DES, the use of 56-Bit Keys, the nature of the DES algorithm, timing attacks, Block cipher design principles, number of rounds, design of function F, key schedule algorithm.

Unit II
Public-Key Cryptography and RSA: Principles of public-key cryptosystems. Public-key cryptosystems. Applications for public-key cryptosystems, requirements for public-key cryptosystems. public-key cryptanalysis. The RSA algorithm, description of the algorithm, computational aspects, the security of RSA: Other Public-Key Cryptosystems: Diffie-hellman key exchange, The algorithm, key exchange protocols, man in the middle attack, Elgamal Cryptographic systems, Elliptic curve arithmetic, abelian groups, elliptic curves over real numbers, elliptic curves over Zp, elliptic curves over GF(2m), Elliptic curve cryptography, Analog of Diffie-hellman key exchange, Elliptic curve encryption/ decryption, security of Elliptic curve cryptography, Pseudorandom number generation based on an asymmetric cipher, PRNG based on RSA.

Unit III
Key Management and Distribution: Symmetric key distribution using Symmetric encryption, A key distribution scenario, Hierarchical key control, session key lifetime, a transparent key control scheme, Decentralized key control, controlling key usage, Symmetric key distribution using asymmetric encryption, simple secret key distribution, secret key distribution with confidentiality and authentication, A hybrid scheme, distribution of public keys, public announcement of public keys, publicly available directory, public key authority, public keys certificates, X-509 certificates. Certificates, X-509 version 3, public key infrastructure. User Authentication: Remote user Authentication principles, Mutual Authentication, one way
Authentication, remote user Authentication using Symmetric encryption, Mutual Authentication, one way Authentication, Kerberos, Motivation, Kerberos version 4, Kerberos version 5, Remote user Authentication using Asymmetric encryption, Mutual Authentication, one way Authentication, federated identity management, identity management, identity federation, personal identity verification.

Unit IV

Unit V
**Reference books:**

**Course Outcomes (COs):**
At the end of the course, the students will be able to:
1. Master information security governance, and related legal and regulatory issues and also master understanding external and internal threats to an organization. (PO 1,3,4)
2. Get familiarity with information security awareness and a clear understanding of its importance and how threats to an organization are discovered, analyzed, and dealt with. (PO 1,3,4)
3. Master fundamentals of secret and public cryptography and master protocols for security services. (PO 1,3,4)
4. Get familiar with network security threats and countermeasures and familiar with network security designs using available secure solutions (such as PGP, SSL, IPSec, etc). (PO 1,3,4)
5. Get familiar with advanced security issues and technologies (such as DDoS attack detection and containment, and anonymous communications,) and exposed to original research in network security and also exposed to the importance of integrating people, processes and technology. (PO 1,3,4)
Digital Forensic and Cyber Crime

Course Code: MCSE20  
Credits: 4:0:0  
Prerequisites: Nil  
Contact Hours: 56  
Course Coordinator/s: Dr Ramani S

Unit 1


Unit II


Unit III


Unit IV


Unit V

Network Forensics: Network Forensic Overview, Performing Live Acquisitions, Developing Standard Procedures for Network Forensics, Using Network Tools. E-mail Investigations: Exploring the Role of E-mail in Investigations, Exploring the Roles of the Client and Server in E-mail, Investigating E-mail Crimes and Violations, Understanding E-mail Servers,
Using Specialized E-mail Forensics Tools. Laboratory Lab exercises using forensic software and Case study data.

**Text Book:**
1. Nelson, Phillips, Frank, Enfinger and Steuart: Computer Forensics and Investigations, Cengage Learning, 2008. Chapters: 1, 2, 4, 6, 7, 8, 9, 10, 11, 12

**Reference Books:**

**Lab Exercises**
The following exercises have to be performed using various open source software tools/utilities mentioned.

**Software Tools:**
- CyberCheck 4.0 - Academic Version
- CyberCheckSuite
- MobileCheck
- Network Session Analyser
- Win-LiFT
- TrueImager
- TrueTraveller
- PhotoExaminer Ver 1.1
- CDRAnalyzer

**Forensics Exercises:**
**I) Disk Forensics:**
1. Identify digital evidences
2. Acquire the evidence
3. Authenticate the evidence
4. Preserve the evidence
5. Analyze the evidence
6. Report the findings

**II) Network Forensics:**
1. Intrusion detection

82
2. Logging (the best way to track down a hacker is to keep vast records of activity on a network with the help of an intrusion detection system)
3. Correlating intrusion detection and logging

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Recall the Indian IT Act 2008 and its amendments. (PO1,3,4)
2. Classify various types of computer crime. (PO1,3,4)
3. Apply computer forensic techniques to identify the digital fingerprints associated with criminal activities. (PO1,3,4)
4. Analyze hidden information from pictures and other files. (PO1,3,4)
5. Apply network forensic tools for network forensic and live data forensic analysis. (PO1,3,4)
Applied Cryptography

Course Code: MCSE21 Credits: 4:0:0
Prerequisites: Contact Hours: 56
Course Coordinator/s: Dr. Mohana Kumar S

Unit I
OSI security architecture: Classical encryption techniques, Cipher principles, Data encryption standard, Block cipher design principles and modes of operation, Evaluation criteria for AES, AES cipher, Triple DES, Placement of encryption function, Traffic confidentiality

Unit II
Key management: Diffie Hellman key exchange, Elliptic curve architecture and cryptography, Introduction to number theory, Confidentiality using symmetric encryption, Public key cryptography and RSA.

Unit III
Authentication requirements: Authentication functions, Message authentication codes, Hash functions, Security of hash functions and MACS, MD5 Message Digest algorithm, Secure hash algorithm, Ripened, HMAC digital signatures, Authentication protocols

Unit IV
Quantum Cryptography and Quantum Teleportation: Heisenberg uncertainty principle, polarization states of photons, quantum cryptography using polarized photons, local vs. non local interactions, entanglements, EPR paradox, Bell’s theorem, Bell basis, teleportation of a single qu-bit theory and experiments.

Unit V
Future trends: Review of recent experimental achievements, study on technological feasibility of a quantum computer candidate physical systems and limitations imposed by noise.
**Text Books:**

**Reference Books:**

**Course Outcomes (COs):**
At the end of the course, students should be able to:
1. Explain the concepts of principles and practice of cryptography and network security. . (PO1,3,4)
2. Present an overview of the Feistel cipher, Distribution of Public Keys, digital signatures and Authentication protocols. . (PO1,3,4)
3. Analyze the security of multiple encryption schemes and Triples DES. . (PO1,3,4)
4. Build secure authentication systems by use of message authentication techniques. . (PO1,3,4)
5. Explain the concepts of principles and practice of visual cryptography. . (PO1,3,4)
Internet of Things (IoT)

Course Code: MCSE22
Credits: 4:0:0
Prerequisites: Nil
Course Coordinator/s: HanumanthaRaju R

Course Contents:

Unit I
Introduction to IoT: What is IoT?, IOT terms and Basic Definitions, Disambiguation of IoT vs IoE vs M2M vs Others, Characteristics of IoT.
Architecture of IoT systems: Things in IoT, Applications of IoT and IoT Reference model, IoT Ecosystem, Enabling Technologies in IoT, Marketplace and Vision of IoT.

Unit II
Hardware aspects of IoT: Sensors and Actuators:
Introduction to Sensors: Workflow of a Sensor in a typical system, Classification of Sensors, Sampling DAC and ADC conversion.
Introduction to Actuators: Workflow of an Actuator in a typical system, Classification of Actuators, Types of Sensors, Interfacing concepts to embedded systems.

Unit III
Communications and networking aspects of IoT:
High bandwidth networking: Ethernet, gigabit Ethernet, Ethernet topologies like bridge and switches, Passive optical fiber network and topologies, WiFi and WiMax. WiFi routers, radius servers, Wireless security with WPA-2, LEAP, enterprise WPA networks
Low Bandwidth Wireless Networks: FSK, LoRa modulation basics, LoRaWAN basics.

Unit IV
Software and middleware aspects of IoT:-
Middleware: Components of Middleware, Types of Databases, Micro
services and API’s.
IP Communication protocols: HTTP, AMQP, MQTT and STOMP.
Protocol definitions, use cases and differences.

Unit V
IoT Platform Design Methodology and Domain Specific IoT.
Futuristic view of IoT, problems pertaining to implementation like scaling,
feasibility and management.

Text Books:

1. Srinivasa K G, Siddesh G.M and HanumanthaRaju R “Internet of

References:

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-
2. Designing the Internet of Things by Adrian McEwenSmart Cities,
Software above the level of a single device,Ebooks on IoT by
O’Reilly
3. Sentilo middleware
4. Mosquito broker
   https://mosquitto.org/
5. Getting started with raspberry pi
   https://www.raspberrypi.org/resources/learn/
6. Arduino basics
7. Wired peripheral protocols
8. OneM2m
   http://www.indiaeulictstandards.in/wp-
   content/uploads/2017/04/oneM2M-for-smart-city-TSDSI-
   presentation-April-21st-2017-Omar-Elloumi.pdf
9. LoRa Modulation
Course Outcomes (COs):

At the end of the course, students should be able to:

1. Understand the basics of IoT.(PO1,3,4)
2. Demonstrate various components, layouts and views in creating IoT applications.(PO1,3,4)
3. Design applications using sensors and actuators.(PO1,3,4)
4. Demonstrate the working of long running tasks in the background using IoT.(PO1,3,4)
5. Demonstrates how to write applications for smart world.(PO1,3,4)
Software Testing

Course Code: MCSE23  Credits: 4:0:0
Prerequisites: SE  Contact Hours: 56
Course Coordinator/s: Pradeep Kumar D

Course Contents:

Unit I

Unit II
Boundary Value Testing, Equivalence Class Testing, Decision Table-Based Testing: Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for the triangle problem, Next Date function, and the commission problem, Guidelines and observations. Decision tables, Test cases for the triangle problem, Next Date function, and the commission problem, Guidelines and observations. Path Testing, Data Flow Testing:DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Definition-Use testing, Slice-based testing, Guidelines and observations.

Unit III
Unit IV

Unit V
Planning and Monitoring the Process, Documenting Analysis and Test: Quality and process, Test and analysis strategies and plans, Risk planning, Monitoring the process, Improving the process, The quality team, Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.

Text Books:


Reference Books:


Course Outcomes (COs):
At the end of the course, a student should be able to:

1. Identify Test cases, Error and fault taxonomies, Levels of testing. (PO1,3,4)
2. Classify different types of testing (Boundary Value Testing, Equivalence Class Testing and Decision Table-Based Testing). (PO1,3,4)

3. Recognize Alternative life - cycle models, recognize Basic concepts for requirements specification, assess context of interaction. (PO1,3,4)

4. Recognize approaches for Test Execution: from test case specifications to test cases, Scaffolding, Generic versus specific scaffolding. (PO1,3,4)

5. Identify and plan strategies to test design specifications document. (PO1,3,4)
Privacy and Security in Online Social Media

Course Code: MCSE24  
Credits: 4:0:0  
Pre-requisites: Nil  
Contact Hours: 56  
Course Coordinator/s: Mrs. Parkavi A

Unit I
What is Online Social Networks, data collection from social networks, challenges, opportunities, and pitfalls in online social networks, APIs.

Unit II
Collecting data from Online Social Media, Trust, credibility, and reputations in social systems.

Unit III
Trust, credibility, and reputations in social systems, Online social Media and Policing.

Unit IV
Information privacy disclosure, revelation and its effects in OSM and online social networks.

Unit V
Phishing in OSM & Identifying fraudulent entities in online social networks.

Reference Links:
Start-up Engineering

Course Code: MCSE25
Prerequisites: Nil
Course Coordinator/s: Prof. Nagabhushana A M

Unit I

Unit II

Unit III
Introduction to HTML/CSS/JS, webpage program, HTTP Request to Rendered Page, Anatomy of web page, HTML: Skeleton and Semantics, CSS, JS, Separation of concerns, Tools , Deployment, DNS, and Custom Domains, Deployment: Dev, Staging, Production, Sidebar: Comparing EC2 vs. local laptops for development, Preliminaries: SSH/HTTP/HTTPS-accessible EC2 dev instance, Creating and managing git branches, Worked Example: Dev, Staging, Production, DNS, Custom Domains, Finding a domain: domize.com, Registering a domain: dnsimple.com, Configuring DNS to work with Heroku, Setting up HTTPS and Google Apps,
Social/Local/Mobile, Virality, Growth, Virality Equation, Local, Local Commerce, Graveyard of Startups.

**Unit IV**


**Unit V**


**Reference Books:**

1. Materials on Startup Engineering, Balaji S. Srinivasan, Stanford University
2. [http://www.wsj.com/articles/SB10001424053111903480904576512250915629460#printMode](http://www.wsj.com/articles/SB10001424053111903480904576512250915629460#printMode)
5. [http://www.nytimes.com/2011/05/08/technology/08class.html?pagewanted=all&_r=0](http://www.nytimes.com/2011/05/08/technology/08class.html?pagewanted=all&_r=0)
7

Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Describing startups and technology trends towards businesses. (PO 1,3,4)
2. Identifying software usage for businesses and ethics. (PO 1,3,4)
3. Figuring out the web usage for business launch. (PO 1,3,4)
4. Assessing the latest technologies for startups. (PO 1,3,4)
5. Preparing software for startups. (PO 1,3,4)
Artificial Intelligence

Course Code: MCSE26 Credits: 4:0:0
Contact Hours: 56

Prerequisites: Knowledge of any advanced programming language, Algorithms and Data structures, Elementary Discrete Mathematics or similar.
Course Coordinator/s: Dr. S. Rajarajeswari and Dr. Annapurna P Patil

Course Contents:

**Unit I**

**Unit II**
Logical Agents: Knowledge-based agents, the wumpus world, Logic, propositional logic, Reasoning patterns in propositional logic, Effective propositional model checking, Agents based on propositional logic First-Order Logic: Representation revisited, Syntax and semantics of first-order logic, using first-order logic, Knowledge engineering in first-order logic. Interference in First-order Logic: Propositional vs first-order inference, Unification and lifting, Forward chaining, Backward chaining, Resolution. Forward chaining, Backward chaining.

**Unit III**
Unit IV


Unit V


Text Books:


Reference Books:

3. http://nptel.ac.in
Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Identify problems that are amenable to specific solution by appropriate AI methods. (PO1,3,4)
2. Utilize various symbolic knowledge representation to specify domains and reasoning tasks of a situated software agent. Use different logical systems for inference over formal domain representations, and trace how a particular inference algorithm works on a given problem specification. (PO1,3,4)
3. Formalize a given problem in the language/framework of different AI methods and solve using basic AI algorithms. (PO1,3,4)
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports. (PO1,3,4)
5. Communicate scientific knowledge at different levels of abstraction in a variety of research settings. (PO1,3,4)
Machine Learning

Course Code: MCSE27
Prerequisites: AI
Course Coordinator/s: Dr. S Rajarajeswari

Credits: 4:0:0
Contact Hours: 56

Course Contents:

Unit I
Introduction
Design a Learning System – Perspectives and Issues in Machine Learning –

Unit II
Linear Models

Unit III
Tree and Probabilistic Models

Unit IV
Dimensionality Reduction and Evolutionary Models

Unit V

Graphical Models

Text Books:


Reference Books:


Course Outcomes (COs):

At the end of the course, the student will be able to:

1. Distinguish between, supervised, unsupervised and semi-supervised learning.(PO1,3,4)
2. Apply the appropriate machine learning strategy for any given problem. (PO1,3,4)
3. Suggest tree and probabilistic learning algorithms for any given problem. (PO1,3,4)
   Modify existing machine learning algorithms to improve classification efficiency. (PO1,3,4)
4. Design systems that use the appropriate graph models of machine learning (PO1,3,4)
Deep Learning

Course Code: MCSE28  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Srinidhi H

Course Contents:

Unit I
Introduction: Human brain, neuron models, neural nets as directed graphs, feedback, neural architectures, knowledge representation, connection to artificial intelligence, Pytorch and Tensorflow.

Unit II
Learning Process: Error-correction learning, memory based learning, Hebbian learning, competitive learning, Boltzmann learning, credit assignment, learning with and without a teacher, learning tasks, memory, statistical learning theory, Backpropagation using MNSIT.

Unit III

Unit IV
Sequence Modelling: Recurrent and recursive nets, practical Methodology, applications. Reinforcement learning.

Unit V

Textbooks:

Course Outcomes (COs):

At the end of the course, student should be able to:

1. Explain the concepts and applications of neural networks and deep learning. (PO1,3,4)
2. Explain how various types of learning work and how they can be used. (PO1,3,4)
3. Apply deep feedforward networks and convolutional to solve practical problems. (PO1,3,4)
4. Demonstrate how recurrent and recursive nets function and how practical problems can be mapped to them. (PO1,3,4)
5. Design end-to-end deep learning architectures involving various types of feedforward networks, auto encoders, RBM, and generative adversarial networks for practical applications. (PO1,3,4)
Social Network Analysis
Course Code: MCSE29 Credits: 4:0:0
Prerequisites: Contact Hours: 56
Course Coordinator/s: Parkavi A

Unit I

Unit II
Network structure. Nodes and edges, network diameter and average path length. Node centralities and ranking on network Node centrality metrics: degree, closeness and betweenness centrality. Eigenvector centrality and PageRank. Algorithm HITS

Unit III

Unit IV

Unit V
Social media mining Natural language processing and sentiment mining. SNA in real world: FB/VK and Twitter analysis Properties of large social networks: friends, connections, likes, re-tweets
Text Books:


Reference Books:


Course Outcomes (COs):

At the end of the course, student should be able to:

1. Know basic notation and terminology used in network science (PO1,3,4)
2. Be able to visualize, summarize and compare networks (PO1,3,4)
3. Understand basic principles behind network analysis algorithms (PO1,3,4)
4. Develop practical skills of network analysis in R programming language (PO1,3,4)
5. Be capable of analyzing real work networks (PO1,3,4)
Natural Language Processing
Course Code: MCSE30 Credits: 4:0:0
Prerequisites: Artificial Intelligence Contact Hours: 56
Course Coordinator/s: Dr. Jagadish S Kallimani

Unit I
Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms; Language, Thought, and Understanding; The State of the Art and The Near-Term Future; Regular Expressions and Automata; Morphology and Finite-State Transducers: Lexicon-free FSTs: The Porter Stemmer, Human Morphological Processing.

Unit II

Unit III

Unit IV

Unit V

Text Book:
1. Daniel Jurafsky and James H Martin, “Speech and Language Processing: An introduction to Natural Language Processing,

Reference Book:

Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Interpret how speech and language technology relies on formal models to capture knowledge, and language processing deals with subparts of words (morphology). (PO 1,3,4)
2. Illustrate the way N-gram tool is used for spelling and pronunciation processing, and part-of-speech tagging mechanism using various categories. (PO 1,3,4)
3. Describe feature structures and unification operation which is used to combine them, and probabilistic parsing to capture more syntactic information. (PO 1,3,4)
4. Outline representations used to bridge the gap from language to commonsense Knowledge (semantic processing), and meanings associated with lexical items. (PO 1,3,4)
5. Emphasize problems that NLP systems face, natural language outputs construction from non-linguistic inputs and machine translation framework approaches. (PO 1,3,4)
Soft Computing

Course Code: MCSE31  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Dr. Jagadish S Kallimani

Unit I
Introduction: Neural networks, Fuzzy logic, Genetic algorithms, Hybrid systems. Artificial Neural Networks: Fundamental concept, Evolution, Basic model of ANN, Important terminologies of ANN, MP neuron, Hebb Network.

Unit II

Unit III

Unit IV

Unit V
Genetic algorithms: Introduction, Basic operations, Traditional algorithms, Simple GA, General genetic algorithms, the schema theorem, Genetic programming, applications.

Reference Books:
1. Principles of Soft computing, S N Sivanandam, Deepa S. N, Wiley, India, (Chapters 1, 2, 3(Up to 3.5), 7, 8, 9, 10, 13, 15 (up to 15.6 & 15.9,15,10).

Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Identify and describe soft computing techniques and their roles in building intelligent machines (PO1,3,4)
2. Identify the components and building block hypothesis of Genetic algorithm. (PO1,3,4)
3. Examine the features of neural network and its applications. (PO1,3,4)
4. Design Genetic algorithm to solve optimization problem. (PO1,3,4)
5. Describe Neuro Fuzzy system for clustering and classification. (PO1,3,4)
Computer System Performance Analysis

Course Code: MCSE32 Credits: 4:0:0
Prerequisites: Probability Theory, matrices, software engineering aspects and queuing theory. Contact Hours: 56
Course Coordinator: Dr. T N R Kumar

Unit I

Unit II

Unit III
Monitors, Program Execution Monitors and Accounting Logs: Monitors: Terminology and classification; Software and hardware monitors, Software versus hardware monitors, Firmware and hybrid monitors, Distributed System Monitors, Program Execution Monitors and Accounting Logs, Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and Interpretation of Accounting log data, Using accounting logs to answer commonly asked questions.

Unit IV
Capacity Planning and Benchmarking: Steps in capacity planning and management; Problems in Capacity Planning; Common Mistakes in Benchmarking; Benchmarking Games; Load Drivers; Remote-Terminal Emulation; Components of an RTE; Limitations of RTEs, Experimental Design and Analysis: Introduction: Terminology, Common mistakes in experiments, Types of experimental designs, $2^k$ Factorial Designs, Concepts, Computation of effects, Sign table method for computing effects; Allocation of variance; General $2^k$ Factorial Designs, General full factorial designs with k factors: Model, Analysis of a General Design, Informal Methods.

Unit V
Queuing Models: Introduction: Queuing Notation; Rules for all Queues; Little’s Law, Types of Stochastic Process. Analysis of Single Queue: Birth-Death Processes; M/M/1 Queue; M/M/m Queue; M/M/m/B Queue with finite buffers; Results for other M/M/1 Queuing Systems. Queuing Networks: Open and Closed Queuing Networks; Product form networks, queuing Network models of Computer Systems. Operational Laws: Utilization Law; Forced Flow Law; Little’s Law; General Response Time Law; Interactive Response Time Law; Bottleneck Analysis; Mean Value Analysis and Related Techniques; Analysis of Open Queuing Networks; Mean Value Analysis;

Text Book:

Reference Books:

Course Outcomes (COs):
At the end of the course, students should be able to:
1. Understand the techniques to approach performance problem and Compare two systems and determine the optimal value of a parameter.(PO1,3,4)
2. Identify performance bottlenecks and characterize the load on a system and Select the number and size of system components and predict the performance of future workloads.(PO1,3,4)
3. Understand the use of different analysis strategies like measurement, simulation, analytical modeling and Implement different techniques in experimental design like factorial design techniques.(PO1,3,4)
4. Understand how to use monitors and accounting logs of systems use to improve the performance of the system and Apply mathematical techniques with stress on learning the types of Queuing models.(PO1,3,4)
5. Apply queuing models to solve problems in computer Networks, Operating system, etc.(PO1,3,4)
Future Skills 2020

Course Code: MCSE33
Prerequisites: Nil
Course Coordinator/s: Prof. Nagabhushana A M

Unit I

Introduction: Current industry overview, Future Skills 2020 research report from IFTF. Sense making: Introduction, VUCA (Volatility, Uncertainty, Complexity and Ambiguity). What is Sense Making? How Sense Making Helps? Steps in sense making, How to do effective sense making? Hurdles in effective sense making. Assignment: A short 1 hour assignment where students will be posed with a situation to exercise their Sense Making ability. It will be assessed at the end of the session.

Unit II

Virtual Collaboration (VC): Introduction, How VC helps? Characteristics of Virtual Collaboration, Types of Virtual Collaboration. Advantages, Disadvantages and Applications of VC. Assignment: The students will be given an assignment applying both the sensemaking skills and Virtual Collaboration skills using the cloud based tools to complete a specific task. This assignment will also cover working in a team using virtual collaboration tools. In order to focus on learning of the specified skills, the end task is kept small and achievable in short time frame.

Unit III

Social Intelligence: Introduction, Hypothesis, Measuring Social Intelligence, Difference between intelligence and Social Intelligence, Derive some of the study done in Social networking theory. Assignment: The assignment will focus on students using their social network to accomplish a specific task.

Unit IV

Crosscultural competency: Introduction, Importance of cross cultural competence in workplace. Nuances of cross cultural differences, Examples to demonstrate the differences. Assignment: Students will have to work with a team member from another culture to complete a specific task.

Unit V

Cognitive Load management: Introduction, Current situation of information overload, Tools and techniques to handle the cognitive load. Importance of these skills in work place. Assignment: Students will be given a specific topic and time to quickly arrive at a good summary of the topic. They will be given access to internet and books to refer. Importance is given to how quickly they can gather, curate and present the summary of the topic.
Reference Books:
1. The detailed report can be found at http://www.iftf.org/uploads/media/SR1382A_UPRI_future_work_skills_s m.pdf
2. The reading material for individual lectures will be shared with the students using Tutor Space.

Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Identify the Future Work skills needed for next 5 years.(PO 1,3,4)
2. Illustrate sense Making Skills through assignments.(PO 1,3,4)
3. Survey the different Virtual Collaboration skills to complete an assignment.(PO 1,3,4)
4. Describe the social intelligence skill and application of the same.(PO 1,3,4)
5. Compose an assignment using Cross-cultural competence and load management skills.(PO 1,3,4)
Cyber Physical Systems

Course Code: MCSE34  Credits: 4:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Dr. Anita Kanavalli

Unit I
Modeling, design, analysis, and implementation of cyber-physical systems

Unit II
Dynamic behavior modeling, state machine composition, and concurrent computation

Unit III
Sensors and actuators. Embedded systems and networks

Unit IV
Feedback control systems, Analysis and verification techniques, temporal logic, and model checking

Unit V
Machine learning topics

Text Book:


Course Outcomes (COs):
At the end of the course, the students will be able to:
1. Apply the afore mentioned cyber-physical systems fundamentals to application domains such as connected and autonomous vehicles, industrial internet, and smart and connected health.
2. Implement cyber-physical systems solutions (e.g., embedded networking protocols, real-time scheduling algorithms, and networked control algorithms).
3. Explore (e.g., survey) cutting-edge research findings in cyber physical systems.
High Performance Computing

Course Code: MCSE35
Prerequisites: Computer Organization
Course Coordinator/s: Mallegowda M

Unit I
Introduction to High–Performance Computers, Memory Hierarchy, CPU Design: Reduced Instruction Set Computers, Multiple–Core Processors, Vector Processors. Parallel Semantics, Distributed Memory Programming.

Unit II

Unit III

Unit IV

Unit V
CUDA Threads: CUDA Thread Organization, Using blockIdx and threadIdx, Synchronization and Transparent Scalability, Thread Assignment, Thread Scheduling and Latency Tolerance. CUDA Memories: Importance of

Text Books:

Reference Books:
3. Michael J. Quin “Parallel Programming in C with MPI and Open MP”, McGraw Hill.

Course Outcomes (COs):
At the end of the course, the student must be able to:
1. Explain the technologies and architectures used for parallel computing (PO-1,2,3)
2. Design and develop parallel programs using Open-MP programming interface. (PO-1,2,3)
3. Elaborate the principles and architecture of message-passing programming paradigm for solving real world problems (PO-1,2,3)
4. Provide an understanding of Graphical Processing Units and their architecture (PO-1,2,3)
5. Analyze the features of GPUs, their functionalities and also Design parallel applications using CUDA-C (PO-1,2,3)
### Annexure

**Table 1**

**Rubrics for Assessment of Student Performance in Laboratory**

<table>
<thead>
<tr>
<th>Trait</th>
<th>Barely Acceptable</th>
<th>Basic</th>
<th>Good</th>
<th>Very Good</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Specifications</strong></td>
<td>The program is producing incorrect results.</td>
<td>The program produces correct results but does not display them correctly.</td>
<td>The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.</td>
<td>The program works and meets all of the specifications.</td>
</tr>
<tr>
<td><strong>Readability</strong></td>
<td>The code is poorly organized and very difficult to read.</td>
<td>The code is readable only by someone who knows what it is supposed to be doing.</td>
<td>The code is fairly easy to read.</td>
<td>The code is exceptionally well organized and very easy to follow.</td>
</tr>
<tr>
<td><strong>Reusability</strong></td>
<td>The code is not organized for reusability.</td>
<td>Some parts of the code could be reused in other</td>
<td>Most of the code could be reused in other programs.</td>
<td>The code could be reused as a</td>
</tr>
<tr>
<td>Documentation</td>
<td>The documentation is simply comments embedded in the code and does not help the reader understand the code.</td>
<td>The documentation is simply comments embedded in the code with some simple header comments separating routines.</td>
<td>The documentation consists of embedded comments and some simple header documentation that is somewhat useful in understanding the code.</td>
<td>The documentation is well written and clearly explains what the code is accomplishing and how.</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delivery</td>
<td>The code was more than 2 weeks overdue.</td>
<td>The code was within 2 weeks of the due date.</td>
<td>The program was delivered within a week of the due date.</td>
<td>The program was delivered on time.</td>
</tr>
<tr>
<td>Efficiency</td>
<td>The code is huge and appears to be patched together.</td>
<td>The code is brute force and unnecessarily long.</td>
<td>The code is fairly efficient without sacrificing readability and understanding.</td>
<td>The code is extremely efficient without sacrificing readability and understanding.</td>
</tr>
</tbody>
</table>
Table 2
Rubrics for assessment of Seminar

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Distinguished (5)</th>
<th>Good (4)</th>
<th>Basic (3)</th>
<th>Unacceptable (1)</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>- Extremely well organized.</td>
<td>- Generally well organized.</td>
<td>- Somewhat organized.</td>
<td>- Poorly organized</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Introduces the purpose of the presentation clearly and creatively.</td>
<td>- Introduces the purpose of the presentation clearly.</td>
<td>- Introduces the purpose of the presentation clearly.</td>
<td>- Does not clearly introduce the purpose of the presentation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Effectively includes smooth, clever transitions which are succinct but not broken up in order to connect key points</td>
<td>- Includes transitions to connect key points and better transitions from idea to idea are noted.</td>
<td>- Includes some transitions to connect key points but there is difficulty in following presentation.</td>
<td>- Uses ineffective transitions that rarely connect points; cannot understand presentation because there is no sequence of information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Student presents information in logical, interesting sequence which audience can follow.</td>
<td>- Most information presented is in logical sequence; A few minor points may be confusing</td>
<td>- Student jumps around topics. Several points are confusing.</td>
<td>- Presentation is broken and disjointed; no apparent logical order of presentation-Ends without a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Ends with an accurate</td>
<td>- Ends with a summary of main</td>
<td>- Ends with a summary or</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Content: Depth and Accuracy</strong></td>
<td>conclusion showing thoughtful, strong evaluation of the evidence presented.</td>
<td>points showing some evaluation of the evidence presented.</td>
<td>conclusion; little evidence of evaluating content based on Evidence.</td>
<td>summary or conclusion.</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>----------------------------------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>-</strong> Speaker provides an accurate and complete explanation of key concepts and theories, drawing upon relevant literature. Applications of theory are included to illuminate issues. -Provides evidence of extensive and valid research on the selected topic, with multiple and varied</td>
<td>-For the most part, explanations of concepts and theories are accurate and complete. Some helpful applications of theory are included. -Presents evidence of valid research on the selected topic, with multiple sources. -Combines existing ideas to form</td>
<td>-Explanations of concepts and/or theories are inaccurate or incomplete. Little attempt is made to tie in theory. There is a great deal of information that is not connected to the current presentation. -Presents evidence of research on the selected</td>
<td>-No reference is made to literature or theory. Presentation is not clear; information that does not support presentation in any way is unnecessarily included. -Presents little or no evidence of valid research on the selected topic. -Shows little evidence of the combination of ideas.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creativity</td>
<td>-Uses the unexpected to full advantage; very original, clever, and creative approach that captures audience's interest.</td>
<td>-Some originality apparent; clever at times; good variety and blending of materials/media.</td>
<td>-Little or no variation; a few original touches but for the most part material presented with little or no creative energy used.</td>
<td>-Bland and predictable. Repetitive with little or no variation; little creative energy used.</td>
<td></td>
</tr>
<tr>
<td>Use of Communication Aids</td>
<td>- Graphics designed reinforce presentation and maximize audience understanding; use of media is varied and appropriate with media not being added simply for the sake of use.</td>
<td>- While graphics relate and aid presentation, media are not as varied and not as well connected to the presentation.</td>
<td>- Occasional use of graphics that rarely support presentation; visual aids were not useful or clear, time wasting use of multimedia;</td>
<td>-- Student uses superfluous graphics, no graphics, or graphics that are so poorly prepared that they detract from the presentation.</td>
<td></td>
</tr>
<tr>
<td><strong>Use of Language</strong></td>
<td>-Poised, clear articulation; proper volume; steady rate; enthusiasm; confidence; speaker is clearly comfortable in front of the group. -Presentation has no misspellings or grammatical errors.</td>
<td>-Clear articulation but not as polished; slightly uncomfortable at times. Most can hear presentation. -Presentation has no more than two misspellings and/or grammatical errors</td>
<td>-Audience occasionally has trouble hearing the presentation; seems uncomfortable. -Presentation has three misspellings and/or grammatical errors.</td>
<td>-Student is anxious and cannot be heard or monotone with little or no expression. -Presentation has four or more spelling errors and/or grammatical errors.</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Eye Contact</strong></td>
<td>-Maintains eye contact; seldom returning to notes; presentation is like a planned conversation.</td>
<td>-Student maintains eye contact most of the time but frequently returns to slides.</td>
<td>-Some eye contact, but not Maintained and at least half the time</td>
<td>-Student reads all or most of slides with no eye contact.</td>
<td></td>
</tr>
<tr>
<td><strong>Viva Voce</strong></td>
<td>- Demonstrates extensive knowledge of the topic by responding confidently, precisely and appropriately to all audience questions.</td>
<td>- Demonstrates knowledge of the topic by responding accurately and Appropriately addressing questions. At ease with answers to all questions but fails to elaborate.</td>
<td>- Demonstrates incomplete knowledge of the topic by responding inaccurately and Inappropriately to questions.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Regularity</th>
<th>Reports to guide regularly for seminar discussion</th>
<th>Reports to guide often for seminar discussion</th>
<th>Does not report to guide for seminar discussion</th>
<th>Has not met the guide at all.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Presentation</td>
<td>Excellent</td>
<td>Good</td>
<td>Average</td>
<td>Poor</td>
</tr>
</tbody>
</table>
Table 3  
Rubrics for assessment of Internship

Deliverables for Student Performance in Internship:

Internship Title: ________________________________

Company Name: ________________________________

Name of Student: _______________________________

Name of Supervisor at Company: __________________

Name of Supervisor at College: ____________________

Each supervisor must fill a rubric for each student:

<table>
<thead>
<tr>
<th>Tools and new Technology Learnt</th>
<th>Basic (0-4 Pts)</th>
<th>Good (5-7 Pts)</th>
<th>Very Good (10 Pts)</th>
<th>Total Possible</th>
<th>Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Few sources at the Industry, aware of quality</td>
<td>Multiple sources of high quality, good judgment of well researched</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
of resources and relevance to tools and Techniques at hand

<table>
<thead>
<tr>
<th>Relevance of the topic chosen to the current market</th>
<th>Fairly Relevant</th>
<th>Moderately Relevant</th>
<th>Highly Relevant</th>
<th>10</th>
</tr>
</thead>
</table>

the information, identification of gaps in knowledge at the Industry and Academics.

and analyzed, continuous efforts at acquiring Information. Identification of the application of the tools and Technology learnt to the present market.
<table>
<thead>
<tr>
<th>Report Writing</th>
<th>Reasonably good organization and lacks clarity in few topics, complete, few omissions,</th>
<th>Sound organization and structure, clear, very few errors, complete, reasonably good style</th>
<th>Excellent organization, no technical or grammar errors, concise and precise, complete</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>grammatically correct, lacks style</td>
<td>documentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demonstration of the Tools Learnt</td>
<td>Moderately be able to demonstrate the tools learnt at the Industry</td>
<td>Efficiently be able to demonstrate the skills learnt and be able to propose an application for the</td>
<td>Excellent demonstration of the tools and techniques learnt and be able to apply it to any</td>
<td>10</td>
</tr>
</tbody>
</table>

128
<p>| <strong>Presentation and viva voce</strong> | Reasonably good communication and presentation, able to give technical answers to some extent | Good, professional communication, good visual aids, able to give technical answers | Excellent professional and technical communication, effective presentations, able to analyze technically and clarify views in viva-voce | 10 |</p>
<table>
<thead>
<tr>
<th>Sl No</th>
<th>Rubrics for assessment of student performance in Project work I</th>
<th>Level of achievement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Excellent (5)</td>
</tr>
<tr>
<td>1</td>
<td><strong>Identificatio</strong>n of Problem Domain and Detailed Analysis</td>
<td>Purpose and need of the project is very well explained.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Study of the Existing Systems / Literature Survey</strong></td>
<td>Existing systems are very well studied. Documents of high standards like IEEE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>3</td>
<td>Objectives of the Proposed Work</td>
<td>All objectives of the proposed work are very well defined.</td>
</tr>
<tr>
<td>4</td>
<td>Design Methodology</td>
<td>Steps to be followed to solve the defined problem are clearly specified. Most suitable design Methodology.</td>
</tr>
<tr>
<td></td>
<td>Tools used for Design</td>
<td>Planning of Project Work</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>5</td>
<td>Clear Understanding</td>
<td>Time frame properly specified and being followed accurately</td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Time frame properly specified and being followed most of the time</td>
</tr>
<tr>
<td></td>
<td>Good</td>
<td>Time frame properly specified, but not being followed</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>Time frame is vaguely specified, and is not followed</td>
</tr>
<tr>
<td></td>
<td>Not Acceptable</td>
<td>Time frame itself is not properly specified</td>
</tr>
<tr>
<td>8</td>
<td>of Presentation</td>
<td>Presentatio n is appropriate but not well arranged. Proper eye contact with audience and clear voice with good spoken language</td>
</tr>
<tr>
<td>---</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>9</td>
<td>Regularity</td>
<td>Student reports to the guide regularly and is consistent in work</td>
</tr>
<tr>
<td><strong>Report of</strong></td>
<td>Project</td>
<td>Project</td>
</tr>
<tr>
<td></td>
<td>Project Phase I</td>
<td>report is according to the specified format References and citations are appropriate and well mentioned</td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
Table 5

Rubrics for assessment of student performance in Project work II

<table>
<thead>
<tr>
<th>Level of achievement</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent (10)</td>
<td></td>
</tr>
<tr>
<td>Good (8)</td>
<td></td>
</tr>
<tr>
<td>Average (6)</td>
<td></td>
</tr>
<tr>
<td>Acceptable (4)</td>
<td></td>
</tr>
<tr>
<td>Unacceptable (2)</td>
<td></td>
</tr>
<tr>
<td>Score</td>
<td></td>
</tr>
<tr>
<td><strong>Incorporation of Suggestions</strong></td>
<td>Changes are made as per modifications suggested during Project Phase I evaluation and new innovations are added</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Project Demonstration</strong></td>
<td>All defined objectives are achieved with some more additional</td>
</tr>
<tr>
<td>Demonstration and Presentation</td>
<td>Contents of Presentation is Appropriate and well arranged Proper eye contact with Satisfactory</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>features. Each Module is working well and is properly demonstrated All modules of project are well integrated and system working is accurate satisfactorily and is properly demonstrated All modules of project are well integrate and system working is satisfactory</td>
</tr>
<tr>
<td>Project Report</td>
<td>Project report is according to the specified format References and citations are appropriate and well mentioned</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Conclusi</td>
<td>Results are</td>
</tr>
<tr>
<td>on and Discussion</td>
<td>presented in very appropriate manner. Project work is well summarized and concluded. Future extensions in the project are very well specified</td>
</tr>
<tr>
<td>Problem formulation</td>
<td>Barely acceptable (0 – 2 Pts)</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td></td>
<td>Bare formulation</td>
</tr>
<tr>
<td></td>
<td>Bare understanding of the problem, with scarce knowledge of relevant material</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each supervisor on the project must fill a rubric for each student.
| Self-motivation and project management | • Slow progress, with barely satisfactory result  
  • Unresponsive to supervisor | • Slow progress, with basic project outcome  
  • Rely on supervisor’s push to work | • Good progress  
  • Need reminder sometimes  
  • Minor problems in project management | • Steady progress  
  • Highly self-motivated  
  Good project management | 5 |
Table 7

**Design Development and Solution Asset Rubric for Project work-II Evaluation**

Project: ____________________________________________
Name of Student: ____________________________________
Name of Supervisor: _________________________________

Each supervisor on the project must fill a rubric for each student

<table>
<thead>
<tr>
<th></th>
<th>Barely acceptable (0 – 2 Pts)</th>
<th>Basic (3 Pts)</th>
<th>Good (4 Pts)</th>
<th>Very Good (5 Pts)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analysis and solving skills</strong></td>
<td>• Obvious solution, sketchy functionalities</td>
<td>• Simple, yet mostly complete solution that solves the stated problem</td>
<td>• Complete solution with nontrivial functionalities that meet the desired needs</td>
<td>• Provide solution to complex problems; Solution optimize desired needs</td>
<td>5</td>
</tr>
</tbody>
</table>
| **Innovation in the Design Solution and self-study** | • Basic concepts used correctly  
• Lack self-study, but apply previously taught technique on a satisfactory level | • Superficial usage of new concepts  
• Self-study of new technique, with basic | • Self-study of new concepts / technique, with good understanding  
• Minor innovative | • New concepts used frequently  
• Self-study of new technique and solve technical difficulties;  
• Innovati | 5 |
<table>
<thead>
<tr>
<th><strong>Self-motivation and project management</strong></th>
<th><strong>understanding</strong></th>
<th><strong>Work</strong></th>
<th><strong>ve work with research value</strong></th>
</tr>
</thead>
</table>
| ● Slow progress, with barely satisfactory result  
● Unresponsive to Supervisor | ● Slow progress, with basic project outcome  
● Rely on supervisor’s push to work | ● Good progress  
● Need reminder sometimes  
● Minor problems in project management | ● Good progress  
● Need reminder sometimes  
● Minor problems in project management |
<p>| 5 |</p>
<table>
<thead>
<tr>
<th></th>
<th>Barely acceptable (0 – 2 Pts)</th>
<th>Basic (3 Pts)</th>
<th>Good (4 Pts)</th>
<th>Very Good (5 Pts)</th>
<th>Total Possible</th>
<th>Total Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content</strong></td>
<td>-Important points covered only superficially</td>
<td>-Covers important points</td>
<td>-All major points covered and explained clearly and correctly</td>
<td>-Major points strongly supported with suitable detail</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-No major errors and misconception</td>
<td>-A few inaccurate or irrelevant points</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Writing</strong></td>
<td>-Frequent errors in spelling and grammar</td>
<td>Some errors in spelling and grammar</td>
<td>-A few errors in spelling and grammar</td>
<td>-Well proofread</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Mostly readable, but a few points are hard to understand</td>
<td>Readable</td>
<td>-Readable and easy to understand</td>
<td>-Clear and easy to understand</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Follow basic written report structure</td>
<td></td>
<td>-Graphs and diagrams used appropriately</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Barely acceptable (0 – 2 Pts)</td>
<td>Basic (3 Pts)</td>
<td>Good (4 Pts)</td>
<td>Very Good (5 Pts)</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>-------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Content</td>
<td>Important points covered only superficially</td>
<td>Covers important points</td>
<td>All major points covered and explained clearly and correctly</td>
<td>Major points strongly supported with suitable detail</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Presentation Skills</td>
<td>Bare organization and preparation</td>
<td>Basic organization and preparation</td>
<td>Good organization and preparation</td>
<td>Excellent organization and preparation Confident and relaxed in the whole presentation Engaging to audience</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>Answer at least</td>
<td>Answer</td>
<td>Answer most</td>
<td>Handle</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>(Q/A)</td>
<td>one questions correctly</td>
<td>most questions correctly</td>
<td>questions correctly and concisely</td>
<td>difficult questions with ease and confidence</td>
<td>-Illustrative explanation</td>
<td></td>
</tr>
</tbody>
</table>