CURRICULUM
for the Academic year 2018 – 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

III & IV SEMESTER B.E

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 prescribed faculty student ratio and achieves excellent academic results. The institute was a participant of the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. All the departments have 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 150 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 good Mentoring/Proctorial system, a fully equipped Sports department, large air-conditioned library with over 1,35,427 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 and all are 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 60th rank in 2018 among the top 100 engineering colleges across India.
About the Department

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

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 stakeholders 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):
A B.E (Computer Science & Engineering) graduate of 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

PROGRAM OUTCOMES (POs):
The Outcomes of the Bachelor of engineering in Computer Science & Engineering Programme are as follows:

Engineering Graduates must be able to:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Understand the principles, architecture and organization of computers, embedded systems and computer networks.

PSO2: Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems that include both hardware and software.

PSO3: Apply software design and development practices to develop software applications in emerging areas such as IoT, Data Analytics, Social Networks, Cloud and High Performance Computing.
### Curriculum Course Credits Distribution

<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>2</td>
<td>9</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Second</td>
<td>2</td>
<td>9</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>Third</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Fourth</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Fifth</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Sixth</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>Seventh</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>26</td>
</tr>
<tr>
<td>Eighth</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>20</td>
<td>2</td>
<td>26</td>
</tr>
</tbody>
</table>

### SCHEME OF TEACHING

#### 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>1</td>
<td>CS31</td>
<td>Engineering Mathematics-III</td>
<td>BS</td>
<td>4 L 0 T P S* 4</td>
<td>56</td>
</tr>
<tr>
<td>2</td>
<td>CS32</td>
<td>Data Structures</td>
<td>PC-C</td>
<td>3 L 1 T P S* 4</td>
<td>42+28</td>
</tr>
<tr>
<td>3</td>
<td>CS33</td>
<td>Discrete Mathematical Structures</td>
<td>PC-C</td>
<td>3 L 0 T P S* 4</td>
<td>42</td>
</tr>
<tr>
<td>4</td>
<td>CS34</td>
<td>Theory of Computation</td>
<td>PC-C</td>
<td>3 L 1 T P S* 4</td>
<td>42+28</td>
</tr>
<tr>
<td>5</td>
<td>CS35</td>
<td>Analog and Digital Design</td>
<td>PC-C</td>
<td>3 L 0 T P S* 4</td>
<td>42+28</td>
</tr>
<tr>
<td>6</td>
<td>CS36</td>
<td>Object Oriented Programming with C++</td>
<td>PC-C</td>
<td>3 L 0 T P S* 3</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>CSL37</td>
<td>Object Oriented Programming Laboratory</td>
<td>PC-C</td>
<td>0 L 0 T P S* 1</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>CSL38</td>
<td>Data Structures Laboratory</td>
<td>PC-C</td>
<td>0 L 0 T P S* 1</td>
<td>28</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>19 L 2 T P S* 25</td>
<td>--</td>
</tr>
</tbody>
</table>
### 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>1.</td>
<td>CS41</td>
<td>Engineering Mathematics-IV</td>
<td>BS</td>
<td>4</td>
<td>56</td>
</tr>
<tr>
<td>2.</td>
<td>CS42</td>
<td>Design and Analysis of Algorithms</td>
<td>PC-C</td>
<td>3 1 0</td>
<td>42+28</td>
</tr>
<tr>
<td>3.</td>
<td>CS43</td>
<td>Microprocessors and Microcontrollers</td>
<td>PC-C</td>
<td>3 0 0 1</td>
<td>42</td>
</tr>
<tr>
<td>4.</td>
<td>CS44</td>
<td>Data Communication</td>
<td>PC-C</td>
<td>4 0 0</td>
<td>56</td>
</tr>
<tr>
<td>5.</td>
<td>CS45</td>
<td>Computer Organization</td>
<td>PC-C</td>
<td>3 0 1</td>
<td>42+28</td>
</tr>
<tr>
<td>6.</td>
<td>CS46-1/2</td>
<td>Unix and Python Scripting/ Unix</td>
<td>PC-C</td>
<td>0 1 2</td>
<td>28+56</td>
</tr>
<tr>
<td>7.</td>
<td>CSL47</td>
<td>Algorithms Laboratory</td>
<td>PC-C</td>
<td>0 0 1</td>
<td>28</td>
</tr>
<tr>
<td>8.</td>
<td>CSL48</td>
<td>Microprocessor Laboratory</td>
<td>PC-C</td>
<td>0 0 1</td>
<td>28</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>1 2 5 1</td>
<td>25</td>
</tr>
</tbody>
</table>

*Self-Study Component:* The topics are included in the syllabus and are considered for evaluation during all the CIE and SEE of that particular subject.

Two credits are allocated to Extra-curricular and Co-curricular Activities (EAC) which will be evaluated during the 8th Semester considering the achievements of a student from semester 1 to 8.
Engineering Mathematics-III

Course Code: CS31                  Credits: 4:0:0:0

Prerequisites: Engineering Mathematics-I and II (MAT101 & MAT201)
Course Coordinator/s: Dr. A Sreevallabha Reddy & Dr. N L Ramesh

Course Contents:

Unit I

**Numerical solution of Algebraic and Transcendental equations:** Method of false position, Newton - Raphson method. **Numerical solution of Ordinary differential equations:** Taylor series method, Euler and modified Euler method, fourth order Runge-Kutta method. **Statistics:** Curve fitting by the method of least squares, fitting a linear curve, fitting a parabola, fitting a Geometric curve, Correlation and Regression.

Unit II

**Linear Algebra I:** Elementary transformations on a matrix, Echelon form of a matrix, rank of a matrix, Consistency of system of linear equations, Gauss elimination and Gauss – Seidal method to solve system of linear equations, eigen values and eigen vectors of a matrix, Rayleigh power method to determine the dominant eigen value of a matrix, diagonalization of a matrix, system of ODEs as matrix differential equations.

Unit III

**Linear Algebra II:** Symmetric matrices, orthogonal diagonalization and Quadratic forms. Linear Transformations, Introduction, Composition of matrix transformations, Rotation about the origin, Dilation, Contraction and Reflection, Kernel and Range, Change of basis.

Unit IV

**Fourier series:** Convergence and divergence of infinite series of positive terms. Periodic functions, Dirchlet conditions, Fourier series of periodic functions of period 2π and arbitrary period, half range Fourier series, Practical harmonic analysis.

Unit V

**Fourier Transforms:** Infinite Fourier transform, Fourier sine and cosine transform, Properties, Inverse transform. **Z-Transforms:** Definition, Standard Z-transforms, Single sided and double sided, Linearity property, Damping rule, Shifting property, Initial and final value theorem, Inverse Z-transform, Application of Z-transform to solve difference equations.

Text Books:

Reference Books:


Course Outcomes (COs):

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

1. Solve the problems of algebraic, transcendental and ordinary differential equations using numerical methods and fit a suitable curve by the method of least squares and determine the lines of regression for a set of statistical data. (PO-1,2, PSO-2)
2. Analyze the concept of rank of a matrix and test the consistency of the system of equations and solution by Gauss Elimination and Gauss Seidel iteration methods. Solve the system of ODE’s by matrix differential equations. (PO-1,2, PSO-2)
3. Do the orthogonal diagonalization and to find the Kernel and Range of Linear transformations. (PO-1,2, PSO-2)
4. Apply the knowledge of Fourier series and expand a given function in both full range and half range values of the variable and obtain the various harmonics of the Fourier series expansion for the given numerical data. (PO-1,2, PSO-2)
5. Evaluate Fourier transforms, Fourier sine and Fourier cosine transforms of functions and apply the knowledge of z-transforms to solve difference equations. (PO-1,2, PSO-2)
Data Structures

Course Code: CS32
Credits: 3:1:0:0
Contact Hours: 42+28

Prerequisites: Fundamentals of Computing
Course Coordinator/s: Mamatha Jadhav V

Course Contents:

Unit I

Unit II
Stacks and Queues: Stacks, Stacks Using Dynamic Arrays, Queues, Circular Queues Using Dynamic Arrays, Evaluation of Expressions, Multiple Stacks and Queues.

Unit III

Unit IV

Unit V

Text Book:

Reference Books:
Course Outcomes (COs):

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

1. Solve real time problems using concepts of dynamic memory allocation, structures and strings. (PO-1,2,3, PSO-2)
2. Differentiate between Stack and Queue variations through numerous applications. (PO-1,2,3, PSO-2)
3. Design programming solutions using variations of linked lists. (PO-1,2,3, PSO-2)
4. Implement hierarchical based solutions using different tree traversal techniques. (PO-1,2,3, PSO-2)
5. Develop solutions for problems based on graphs. (PO-1,2,3, PSO-2)
Discrete Mathematical Structures

Course Code: CS33 Credits: 3:0:0:1
Prerequisites: Basic Mathematics
Course Coordinator/s: Srinidhi H

Course Contents:

Unit I

Unit II

Unit III

Unit IV
Graph Theory: Introduction to Graph theory- Definitions, sub graphs, complements, and graph isomorphism, Euler’s trails and circuits, Hamilton paths and Cycles. Planar graphs, Euler’s Theorem, Graph Coloring. Self-Study: Applications of Hamilton Circuits.

Unit V

Self-Study Evaluation:
- The topics are integral part of the course.
- No formal lectures will be held for the self-study topics.
- The course co-ordinator may provide reading materials for self-study topics (optional).
- The topics prescribed under self-study in curriculum are part of CIE and SEE.
Text Books:


Reference Book:


Course Outcomes (COs):

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

1. Understand the notion of mathematical logic & proofs and be able to apply them in problem solving. (PO-1,2,4,PSO-2,3)
2. Solve problems which involve discrete data structures such as relations and discrete functions (PO-1,2,4,5,10, PSO-2)
3. Apply basic counting techniques and combinatorics in the context of discrete probability. (PO-1,4,5,10, PSO-2)
4. Demonstrate knowledge of fundamental concepts in graphs, (PO-1,2, 5,10, PSO-2)
5. Demonstrate knowledge of trees and its properties using various modeling techniques. (PO-1, 2, 5, 10, PSO-2)
Theory of Computation

Course Code: CS34
Credits: 3:1:0:0

Prerequisites: Basic Mathematics
Course Coordinator/s: D.S. Jayalakshmi

Course Contents:

Unit I
Introduction to Finite Automata, structural representations, automata and complexity, the central concepts of automata theory, deterministic finite automata, nondeterministic finite automata, an application of finite automata, finite automata with epsilon transitions.

Unit II
Regular expressions, finite automata and regular expressions, applications of regular expressions, proving languages not to be regular, closure properties of regular languages, equivalence and minimization of automata.

Unit III
Context–free grammars, parse trees, applications, ambiguity in grammars and languages, definition of the pushdown automata, the languages of a PDA, equivalence of PDAs and CFGs.

Unit IV
Deterministic Pushdown Automata, normal forms for CFGs, the pumping lemma for CFGs, closure properties of CFLs.

Unit V
The Turing machine, programming techniques for Turing Machines, extensions to the basic Turing machine, restricted Turing machines, Turing machine and computers, Un decidability: A language that is not recursively enumerable, an undecidable problem that is RE, definition of Post’s Correspondence problem.

Text Book:

Reference Books:
Course Outcomes (COs):

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

1. Explain the basic concepts of formal languages and finite automata. (PO-1, PSO-2)
2. Construct automata to accept strings from a specified language. (PO-1,2, 3, PSO-2)
3. Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, between PDAs, CFGs and normal forms of CFGs. (PO-1,2,3, PSO-2)
4. Prove the various closure and decision properties of formal languages. (PO-1,2, PSO-2)
5. Explain the concepts of Un decidability, RE languages and Post Correspondence problem. (PO-1,2, PSO-2)
Analog & Digital Design
Course Code: CS35
Credits: 3:0:1:0
Contact Hours: 42+28
Prerequisites: Basic Electronics
Course Coordinator/s: Veena G.S & Aparna R

Course Contents:

Unit I

Unit II
Wave shaping circuits: Basic RC low pass circuits, RC low pass circuit as integrator, Basic RC high pass circuit, RC high pass as differentiator. Tabular method for minimization of Boolean functions, Combinational circuits: Half adder, full adder(realization using NAND gates), adder-sub tractor unit, ripple and fast adders, multiplexers, decoders, encoders, code converters, arithmetic comparison circuits.

Unit III
Wave shaping circuits, diode clipper circuits , diode clamper circuit, integrated circuit multivibrators using 555 (Timer IC) (astable, monostable circuits). Basic latch, gated SR latch, gated D latch, T FF, JK FF, truth table, characteristics equation and excitation tables of all the four types of FFs. Registers: Shift registers, parallel access registers.

Unit IV
Feedback amplifiers: Classification of amplifiers, amplifiers with negative feedback, and advantages of negative feedback. Series and shunt linear regulators, linear IC voltage regulators. Study of asynchronous counters: Up, down counters, reset synchronization, decade counter, Ring counter, Johnson counter, truncated counters.

Unit V
Study of synchronous sequential circuits: Basic design steps, Mealy state model, Mealy type FSM for serial adder. Design of a counter using sequential circuits approach using different FFs for different modulo values and design of random counters.

Text Books:
Reference Books:


Course Outcomes (COs):

At the end of the course, the student will be able to:
1. Describe the working of different analog circuits like op-amp, wave shaping, feedback amplifiers, and regulator circuits, digital combinational and sequential circuits. (PO-1,2, PSO-2)
2. Explain the various techniques used for Boolean function minimization. (PO-1,2, PSO-2)
3. Construct analog wave shaping circuits and digital ALU circuits. (PO-2,5, PSO-2)
4. Examine the characteristics of flip flops and amplifier circuits. (PO-2, PSO-2)
5. Design asynchronous and synchronous sequential circuits. (PO1, PO-2,5, PSO-2)

Laboratory Exercises

• Experiments will be conducted in the Laboratory using Hardware and using Simulation Software like Multisim
• Experiments will be designed for Analog topics-Op-amps, Wave Shaping circuits, Amplifiers and Regulators
• Experiments will be designed for Digital topics- Basic digital design, combinational and sequential circuits
Object Oriented Programming with C++

Course Code: CS36

Credits: 3:0:0:0

Contact Hours: 42

Prerequisites: Nil

Course Coordinator/s: Chethan Shetty

Course Contents:

Unit I
Introduction: Overview of C++, Sample C++ program, Different data types, operators, expressions, and statements, arrays and strings, pointers & function components, recursive functions, user -defined types, function overloading, inline functions, Classes & Objects – I: classes, Scope resolution operator, passing objects as arguments, returning objects, and object assignment.

Unit II
Classes & Objects –II: Constructors, Destructors, friend functions, Parameterized constructors, Static data members, Functions, Arrays of objects, Pointers to objects, this pointer, and reference parameter, Dynamic allocation of objects, Copy constructors, Operator overloading using friend functions such as +, - , pre-increment, post-increment, [ ] ,(), , >, overloading <<,>>.

Unit III
Templates: Generic functions and Generic classes, Inheritance : Base Class, Inheritance and protected members, Protected base class inheritance, Inheriting multiple base classes, Constructors, Destructors and Inheritance, Passing parameters to base class constructors, Granting access, Virtual base classes.

Unit IV
Virtual functions, Polymorphism: Virtual function, calling a Virtual function through a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, pure virtual functions, Abstract classes, Using virtual functions, Early and late binding.

Unit V
Exception Handling, I/O System Basics, File I/O: Exception handling fundamentals, Exception handling options, C++ stream classes, Formatted I/O, fstream and the File classes, Opening and closing a file, Reading and writing text files.

Text Book:

2. Mark priestley.
Reference Books:


Course Outcomes (COs):

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

1. Identify classes, objects, members of a class and the relationships among them needed to solve a specific problem. (PO-1,2,3, PSO-2)
2. Demonstrate the concept of constructors and destructors. And create new definitions for some of the operators (PO-1,2,3, PSO-2)
3. Create function templates, overload function templates, Understand and demonstrate the concept data encapsulation and inheritance (PO-1,2,3, PSO-2)
4. Demonstrate the concept of polymorphism with virtual functions. (PO-1,2,3, PSO-2)
5. Demonstrate the concept of file operations, streams in C++ and various I/O manipulators. (PO-1,2,3, PSO-2)
Object Oriented Programming Laboratory

Course Code: CSL37
Credits: 0:0:1:0
Contact Hours: 28

Prerequisites: Nil
Course Coordinator/s: Chethan Shetty

Course Contents:

1. Inline functions & function overloading.
2. Classes & objects.
3. Constructors, destructors & static data members.
4. Friend functions & generic functions.
5. Operator overloading.
6. Inheritance - protected members, protected base class inheritance, inheriting multiple base classes.
7. Passing parameters to base class constructors, granting access and virtual base class.
8. Virtual functions and polymorphism.
9. Pure virtual functions and abstract classes.
10. Standard Template Library (STL).
11. Exception handling.

Reference Books:


Course Outcomes (COs):

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

1. Develop classes incorporating object-oriented techniques. PO-1,2,3, PSO-2)
2. Design and implement object-oriented concepts of inheritance and polymorphism. PO-1,2,3, PSO-2)
3. Illustrate and implement STL class of containers and need for exceptions to handle errors for object oriented programs. PO-1,2,3, PSO-2)
Data Structures Laboratory

Course Code: CSL38  Credits: 0:0:1:0
Prerequisites: Nil  Contact Hours: 28

Course Coordinator/s: Mamatha Jadhav V

Course Contents:
1. Illustrating Pointers for data operations
2. Examining Dynamic memory allocations
3. Composing Arrays in programs
4. Managing Structures in applications
5. Organizing Stacks in programs
6. Constructing Queues for applications
7. Setting up Linked lists for data set operations
8. Formulating Trees for data set maintenance
9. Developing applications to solve Graph based problems

Text Book:

Reference Book:

Course Outcomes (COs):

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

1. Develop programming solutions for real time problems using dynamic memory allocation, structures and strings. (PO-1,2,3, PSO-2)
2. Develop programming solutions for real time applications using stack, Queues and linked lists (PO-1,2,3, PSO-2)
3. Design hierarchical based programming solutions using different tree traversal techniques and graph theory (PO-1,2,3, PSO-2)
Engineering Mathematics-IV

Course Code: CS41

Credits: 4:0:0:0

Contact Hours: 56

Prerequisites: Engineering Mathematics-I and II (MAT101 & MAT201)

Course Coordinator/s: Dr. A Sreevallabha Reddy & Dr. N L Ramesh

Course Contents:

Unit I

**Finite Differences and Interpolation:** Forward, Backward differences, Interpolation, Newton-Gregory Forward and Backward Interpolation, formulae, Lagrange interpolation formula and Newton divided difference interpolation formula (no proof).

**Numerical Differentiation and Numerical Integration:** Derivatives using Newton-Gregory forward and backward interpolation formulae, Newton-Cotes quadrature formula, Trapezoidal rule, Simpson 1/3rd rule, Simpson 3/8th rule.

Unit II

**Random Variables:** Random Variables (Discrete and Continuous), Probability density function, Cumulative distribution function, Mean, Variance, Moment generating function.

**Probability Distributions:** Binomial distribution, Poisson distribution, Normal distribution, Exponential distribution and Uniform distribution.

Unit-III

**Joint probability distribution:** Joint probability distribution (both discrete and continuous), Conditional expectation, Simulation of random variable.

**Stochastic Processes:** Introduction, Classification of stochastic processes, discrete time processes, Stationary, Ergodicity, Autocorrelation, Power spectral density.

Unit IV

**Markov Chain:** Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states, Markov and Poisson processes.

**Queuing theory:** Introduction, Concepts and M/G/1 and M/M/1 queuing systems with numerical illustration.

Unit V

**Sampling Theory:** Sampling, Sampling distributions, Standard error, Weak law of large numbers(without proof), Central limit theorem, Basics of parametric estimation, Test of Hypothesis for means, Confidence limits for means, Z- test, Student’s t-distribution, F-distribution, Chi-Square distribution as a test of goodness of fit.

Text Books:

Reference Books:


Course Outcomes (COs):

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

1. Use a given data for equal and unequal intervals to find a polynomial function for estimation. Compute maxima, minima, curvature, radius of curvature, arc length, area, surface area and volume using numerical differentiation and integration. Also solve partial differential equations analytically. (PO-1,2, PSO-2)
2. Apply the concept of probability distribution to solve engineering problems. (PO-1,2, PSO-2)
3. Apply the concept of Joint distribution of random variables to understand the basic concepts of stochastic process. (PO-1,2, PSO-2)
4. Apply the concept of Markov Chain in prediction of future events and in queuing models. (PO-1,2, PSO-2)
5. Use the concepts of sampling to make decision about the hypothesis. (PO-1,2, PSO-2)
Design and Analysis of Algorithms

Course Code: CS42
Credits: 3:1:0:0

Prerequisites: Data Structures & Programming Language

Course Coordinator/s: Pramod C Sunagar

Course Contents:

Unit I

Unit II

Unit III

Unit IV
Unit V

**NP and Computational Intractability:** Polynomial-Time Reductions A First Reduction: Independent Set and Vertex Cover, Reducing to a More General Case: Vertex Cover to Set Cover, NP-Complete Problems: Circuit Satisfiability: A First NP-Complete Problem, General Strategy for Proving New Problems NP-Complete, Sequencing Problems: The Traveling Salesman Problem, The Hamiltonian Cycle Problem.

**Text Book:**


**Reference Book:**


**Course Outcomes (COs):**

At the end of the course, the student will be able to:
1. Define the basic concepts and analyze worst-case running times of algorithms using asymptotic analysis.(PO-1,2,PSO-2)
2. Recognize the design techniques for graph traversal using representative algorithms.(PO-1,2,3,PSO-3)
3. Identify how divide and conquer works and analyze complexity of divide and conquer methods by solving recurrence.(PO-1,2,3,PSO-3)
4. Illustrate Greedy paradigm and Dynamic programming paradigm using representative algorithms.(PO-1,2,3,4,PSO-2,3)
5. Describe the classes P, NP, and NP-Complete and be able to prove that a certain problem is NP-Complete.(PO-1,2,3,4,PSO-2,3)
Microprocessors and Microcontrollers

Course Code: CS43
Credits: 3:0:0:1
Contact Hours: 42
Prerequisites: Nil
Course Coordinator/s: Hanumantharaju R

Course Contents:

Unit I
An Introduction to Microprocessor: Processor architecture and organization, Abstraction in hardware design, MU0 - a simple processor, Instruction set design, Processor design tradeoffs, The Reduced Instruction Set Computer, Cortex-M0 Technical Overview, implementation Features.
Self-Study: System Features, Debug Features, Advantages.

Unit II
Programming model, Operation Modes and States, Architecture, Registers and Special Registers Behaviors of the Application Program Status Register (APSR), Memory System Overview. Introduction to Cortex-M0 Programming.
Self-Study: Stack Memory Operations.

Unit III
Instruction Set, Instruction Usage Examples, implementation of various structures like loop, switch, functions, and subroutines.
Self-Study: Programs with subroutines.

Unit IV

Unit V
Sensors, Thermistor, LDRs, LEDs 7 segment, LCD, Stepper motor, relays, Actuator, and ADCs.

Self-Study Evaluation:
• The topics are integral part of the course.
• No formal lectures will be held for the self-study topics.
• The course co-ordinator may provide reading materials for self-study topics (optional).
• The topics prescribed under self-study in curriculum are part of CIE and SEE.
Text Books:


Course Outcomes (COs):

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

1. Explain the architecture and organization of the Processors. (PO-2, 3, 4, PSO-1)
2. Design Operation Modes and States using special purpose registers. (PO-1,2, 3, 4,PSO-1)
3. Evaluate cortex M0 in assembly instructions and write embedded C programs using CMSIS features. (PO-3,4,5,12,PSO-2)
4. Devise programs using interrupt capabilities (PO-2,4,5,PSO-2)
5. Compare the working of various sensors and actuators and their interface with microcontrollers (PO-2,4,12,PSO-1)
Data Communication

Course Code: CS44  Credits: 4:0:0:0
Prerequisites: Nil  Contact Hours: 56
Course Coordinator/s: Sanjeetha R

Course Contents:

Unit I

Unit II

Unit III

Unit IV

Unit V

Text Book:

Reference Books:
Course Outcomes (COs):

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

1. Identify the different types of network topologies and protocol models. (PO-1, 2, 3, 4, PSO-I).
2. Differentiate between digital and analog transmissions. (PO-1, 2, 3, 4, PSO-I).
3. Solve problems in error detection and correction at data link layer. (PO-1, 2, 3, 4, PSO-I).
4. Discriminate between different access control methods to shared transmission media. (PO-1, 2, 3, 4, PSO-I).
5. Compare the working of wired and wireless networks. (PO-1, 2, 3, 4, PSO-I).
Computer Organization

Course Code: CS45

Credits: 3:0:1:0

Contact Hours: 42+28

Prerequisites: Nil

Course Coordinator/s: Chandrika Prasad

Course Contents:

Unit I

Language of the Computer: Operation of the computer hardware, Operands of the Computer Hardware, Signed and Unsigned numbers, Representing Instructions in the Computer, Logical Operations, Instructions for making Decisions, Supporting procedures in the computer hardware, Communicating with people: ASCII versus Binary numbers, compiling a string copy procedure, showing how to use C strings, ARM addressing for 32-bit immediate and more complex addressing modes.

Unit II

Arithmetic unit: Multiplication of two numbers, A signed operand multiplication, Booth algorithm, Bit pair recoding and CSA – integer division, IEEE standard for floating point numbers, Operations, Guard bits and truncation.

Unit III


Unit IV

Memory unit: Introduction, The basics of Caches: Accessing a cache, Handling cache misses, Handling writes, Designing the memory system to support caches, Measuring and improving cache performance: Reducing cache misses by more flexible placement of blocks, Locating a block in the cache, choosing which block to replace, Reducing the miss penalty using multilevel caches, Virtual memory: Placing a page and finding it again, Page faults, TLB.

Unit V

Input Output Unit: Introduction, Dependability, Reliability, and availability, Disk storage, Flash memory, Connecting processors, memory, and I/O devices, Interfacing I/O devices to the processor, memory, and operating system, I/O performance measures, Designing an I/O system, parallelism and I/O, Real stuff: Sun Fire x4150 server.

Text Books:

Reference Books:


Course Outcomes (COs):

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

1. Identify ARM instruction set, ARM instruction format and ARM addressing modes. (PO-1,5, PSO-1,2)
2. Appraise different algorithms used to perform fast multiplication and division also represent the floating-point number in IEEE format. (PO-1,2,5, PSO-1, 2)
3. Design a datapath for MIPS architecture and understand the importance of pipelining. (PO-1,3,5, PSO-1,2)
4. Assess the cache memory performance and also recognize the advantages of using virtual memory technique. (PO-1,2,3,5, PSO-1,2)
5. Demonstrate Interfacing of I/O devices with processor memory and operating system including data transfer between memory and I/ devices. (PO-1, 3, 5, PSO-1,2)

List of Lab Exercises:

1. Demonstrating instruction execution stages using MarieSim Simulator.
2. Executing an ARM programs using ARMsim simulator.
3. Designing an ALU to perform various functions using Logisim simulator.
4. Implementing different multiplication algorithms using Logisim simulator.
5. Implementing pipeline technique using CPUOS simulator.
6. Executing MIPS programs using QtSpim simulator.
7. Designing memory system operations using Logisim simulator.
UNIX and Python Scripting

Course Code: CS46-1
Credits: 0:1:2:0
Contact Hours: 28+56
Prerequisites: Nil
Course Coordinator/s: J Geetha

Course Contents:

1. **Introduction to Unix and Shell programming:** Unix, Unix Architecture, Unix Standardization, Test Macros.
3. **Filters and Filters Using Regular Expressions:** The Sample Database, pr, head, tail, cut, paste, sort, uniq, tr, grep, Basic Regular Expressions, Extended Regular Expressions and egrep, sed, Line Addressing, Text Editing.
4. **Introduction to Python:** Hello Python, Working with Text, Designing and Using Functions, Repeating Code Using Loops.
5. **Storing Collections of Data:** Lists, Sets, Tuples, Dictionaries.
6. **Reading and Writing Files:** Opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, Multiline Records.
8. **Creating Graphical User Interfaces:** Building a Basic GUI using Tkinter, Object oriented GUI.
9. Python Database.

Text Books:


Reference Books:

Course Outcomes (COs):

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

1. Utilise the UNIX environment effectively to perform a range of system-level tasks (PO-2, 5, 12, PSO-1)
2. Analyse, write and apply shell scripts to solve system-level problems (PO-1, 2, 3, 5, PSO-2)
3. Understand the various data structures available in Python programming language and apply them in solving computational problems (PO-1, 2, 3, 5, PSO-2)
4. Apply common built-in Python language functions to solve problems (PO-1, 2, 3, 5, PSO-2)
5. Construct modular Python applications that serve as a functional component of a larger collaborative programming project (PO-1, 2, 3, 5, 12, PSO-3)
UNIX and Ruby on Rails

Course Code: CS46-2
Credits: 0:1:2:0
Contact Hours: 28+56

Prerequisites: Nil
Course Coordinator/s: J Geetha

Course Contents:
1. **Introduction to Unix and Shell programming:** Unix, Unix Architecture, Unix Standardization, Test Macros.
3. **Filters and Filters Using Regular Expressions:** The Sample Database, pr, head, tail, cut, paste, sort, uniq, tr, grep, Basic Regular Expressions, Extended Regular Expressions and egrep, sed, Line Addressing, Text Editing.
4. **Introduction to Ruby:** Installing Ruby, Ruby Is an Object-Oriented Language, Arrays and Hashes, Control Structures, Regular Expressions, Blocks and Iterators.
5. **Classes and Containers:** Objects and Attributes, Classes Working with Other Classes, Access Control, Variables, Arrays, Hashes, Blocks and Iterators.
6. **Sharing Functionality, Standard Types, Regular Expressions:** Inheritance and Messages, Modules, Mixins, Iterators and the Enumerable Module, Ruby’s Regular Expressions, Advanced Regular Expressions.
8. **Advanced Rails:** Creating the Blog Application, Creating the Project Databases, creating a Database Table, generating a Controller, Up and Running with Scaffolding.

Text Books:


Reference Books:

Course Outcomes (COs):

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

1. Utilise the UNIX environment effectively to perform a range of system-level tasks (PO- 2, 5, 12, PSO-1)
2. Analyse, write and apply shell scripts to solve system-level problems (PO-1, 2, 3, 4, 5, PSO-2)
3. Understand the various data structures available in Ruby programming language and apply them in solving computational problems (PO-1, 2, 3, 5, PSO-2)
4. Apply common built-in Ruby language functions to solve problems (PO-1, 2, 3, 5, PSO-2)
5. Construct web applications using Rails with Ruby programming (PO-1, 2, 3, 5, 12, PSO-3)
Algorithms Laboratory

Course Code: CSL47          Credits: 0:0:1:0
Prerequisites: Data Structures, C++, C
Course Coordinator/s: Pramod C Sunagar

Course Contents:

There shall be a minimum of 2 exercises conducted on each of the following topics:
1. Asymptotic bounds and functions.
2. Sorting techniques.
3. Stable matching.
5. Graph traversal techniques.

Reference Books:


Course Outcomes (COs):

At the end of the course, the student will be able to:
1. Define the basic concepts and analyze worst-case running times of algorithms using asymptotic analysis.(PO-1,4,10,PSO-2)
2. Recognize the design techniques for graph traversal, divide and conquer, greedy and dynamic programming paradigm using representative algorithms.(PO-1,2,3, 5, 7, 9, 10, PSO-3)
3. Illustrate Branch and bound paradigm through NP complete problems.( PO-1, 3, 4, 10,12, PSO-2,3)
Microprocessor Laboratory

Course Code: CSL48
Credits: 0:0:1:0
Contact Hours: 28

Prerequisites: Nil
Course Coordinator/s: Veena G S

Course Contents:

1. Introduction to the Architecture of Cortex M0 Processor.
2. Devise the need of Cross compiler(Keil) with assembly programs.
3. Assembly programs using Logical instructions, Shift and Rotate instructions.
5. Assembly programs demonstrating the usage of switch, Extracting/Clearing bits.
6. Assembly programs demonstrating the usage of stack operations and functions.
7. Introduction to NuMicro MCU Learning Board.
8. Interfacing programs on LED and Buzzer.
9. Interfacing programs on LCD and Seven Segment display.
10. Interfacing programs on LCD and interrupt programs.
11. Interfacing programs with ADC.
12. Programs on Self study components.

Text Book:


Course Outcomes (COs):

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

1. Devise M0 processor assembly programs using searching and sorting techniques. (PO-1,2,4,5,PSO-1)
2. Test assembly language programs with I/O devices connected to M0 processor. (PO-1,2,4,5, PSO-2)
3. Develop interrupt driven programs using hardware and software interrupt methods of M0 processor. (PO-1,2,4, PSO-2)