RAMAIAH
Institute of Technology

CURRICULUM
for the Academic year 2020 – 2021

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's 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 304 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 64th rank in 2019 among the top 100 engineering colleges across India.

About the Department

Year of Establishment	1984
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, 15 faculty with the doctorate degree and 15pursuing 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

1. Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization

2. Establishing research clusters in emerging areas in collaboration with globally reputed organizations

3. 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 promotes innovation towards betterment of the society

MISSION OF THE DEPARTMENT

1. To produce Computer Science graduates who, trained in design and implementation of computational systems through competitive curriculum and research in collaboration with industry and research organizations.

2. To educate students in technology competencies by providing professionally committed faculty and staff.

3. 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 teamwork: 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

Semester	Humanities & Social Sciences (HSS)	Basic Sciences/ Lab (BS)	Engineering Sciences/ Lab (ES)	Professional Courses - Core (Hard core, soft core, Lab) (PC-C)	Professional Courses-Electives (PC-E)	Other Electives (OE)	Project Work/ Internship (PW /IN)	Extra & Co-curricular activities (EAC)	Total credits in a semester
First		9	11						20
Second	2	8	10						20
Third		4	3	18					25
Fourth		7		18					25
Fifth	3			15	3	3			24
Sixth				11	3	3	4		21
Seventh	3			10	6			1	20
Eighth					3		14	3	20
Total	8	28	24	72	15	6	18	4	175

SCHEME OF TEACHING

III Semester

SI. No	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	CS31	Engineering Mathematics-III	BSC	3	1	0	4	42+28
2.	CS32	Data Structures	PCC	3	1	0	4	42+28
3.	CS33	Discrete Mathematical Structures	PCC	4	0	0	4	56
4.	CS34	Theory of Computation	PCC	3	1	0	4	42+28
5.	CS35	Analog and Digital Design	PCC	3	0	1	4	42+28
6.	CS36	Object Oriented Programming	PCC	3	0	0	3	42
7.	CSL37	Object Oriented Programming Laboratory	PCC	0	0	1	1	28
8.	CSL38	Data Structures Laboratory	PCC	0	0	1	1	28
Total				19	3	3	25	--

Note: Use Java for OOP

IV Semester

SI. No	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	CS41	Engineering Mathematics-IV	BSC	3	1	0	4	42+28
2.	CS42	Design and Analysis of Algorithms	PCC	3	1	0	4	42+28
3.	CS43	Microprocessors and Microcontrollers	PCC	3	0	1	4	42+28
4.	CS44	Data Communication & Networking	PCC	4	0	0	4	56
5.	CS45	Computer Organization and Architecture	PCC	3	1	0	4	42+28
6.	CS46	Software Engineering	PCC	3	0	0	3	42
7.	CSL47	Algorithms Laboratory	PCC	0	0	1	1	28
8.	CSL48	Python Programming Laboratory	PCC	0	0	1	1	28
Total			Total	19	3	3	25	--

Note: Use Java for Algorithms Lab

Engineering Mathematics-III

Course Code:CS31 **Course Credits:**3:1:0
Prerequisite: Calculus **Contact Hours:**42+28
Course Coordinators: Dr. N. L. Ramesh & Dr. A. Sreevallabha Reddy

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 linear, quadratic and geometric curves, 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 – Seidel 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, Solution of system of ODEs using matrix method.

Unit-III

Linear Algebra II: Symmetric matrices, Orthogonal diagonalization and Quadratic forms, Vector Spaces, Linear combination and span, Linearly independent and dependent vectors, Basis and Dimension, Linear transformations, 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, Dirichlet's 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.

Textbooks:

1. Erwin Kreyszig-Advanced Engineering Mathematics-Wiley-India publishers- 10^{th} edition-2015.
2. B.S.Grewal - Higher Engineering Mathematics - Khanna Publishers – 44^{th} edition-2017.

Reference Books:

1. David C. Lay – Linear Algebra and its Applications – Jones and Bartlett Press – 3rd edition – 2011.
2. Peter V. O'Neil – Advanced Engineering Mathematics – Thomson Brooks/Cole – 7th edition – 2011.
3. Gareth Williams – Linear Algebra with Applications, Jones and Bartlett Press – 8rd edition – 2012.

Course Outcomes (COs):

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

1. Apply numerical techniques to solve engineering problems and fit a least squares curve to the given data. (PO-1,2 PSO-2)
2. Test the system of linear equations for consistency and solve system of ODE's using matrix method. (PO-1,2 PSO-2)
3. Diagonalize a given matrix orthogonally and find kernel and range of linear transformation. (PO-1, 2 PSO-2)
4. Construct the Fourier series expansion of a function/tabulated data. (PO-1,2 PSO-2)
5. Evaluate Fourier transforms and use Z-transforms to solve difference equations. (PO-1, 2 PSO-2)

Data Structures

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

Prerequisites: Fundamentals of Computing
Course Coordinator/s: Vandana Sudhakar Sardar

Course Contents:

Unit I

Basic Concepts: Pointers and Dynamic Memory Allocation, Algorithm Specification, Data Abstraction. Arrays and Structures: Arrays, Dynamically Allocated Arrays, Structures and Unions, Polynomials, Sparse Matrices, Representation of Multidimensional Arrays, Strings.

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

Linked Lists: Singly Linked lists and Chains, Representing Chains in C, Linked Stacks and Queues, Polynomials, Additional List operations, Sparse Matrices, Doubly Linked Lists.

Unit IV

Trees: Introduction, Binary Trees, Binary Tree Traversals, Additional Binary Tree Operations, Threaded Binary Trees, Heaps, Binary Search Trees, Selection Trees, Forests, Representation of Disjoint Sets, Counting Binary Trees.

Unit V

Graphs: The Graph Abstract Data Type, Elementary Graph Operations. Priority Queues: Single- and Double-Ended Priority Queues, Leftist Trees. Efficient Binary Search Trees: AVL Trees.

Textbook:

1. Horowitz, Sahni, Anderson-Freed: Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2008.

Reference Books:

1. Yedidyah, Augenstein, Tannenbaum: Data Structures Using C and C++, 2nd Edition, Pearson Education, 2003.
2. Data Structures, SeynourLipschutz and GAV Pai, Schaum's Outlines, McGraw Hill, 2008.
3. Richard F. Gilberg and Behrouz A. Forouzan: Data Structures- A Pseudocode Approach with C, Cengage Learning, 2005

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. Implement storage and retrieval of ordered data using stacks and queues as well as select appropriate data structures as applied to specified problem definition (PO-1,2,3, PSO-2)
3. Implement dynamic storage, retrieval and search operations of unordered data using linked list and its variants. (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: 4:0:0**
Contact Hours: 56

Prerequisites: Basic Mathematics
Course Coordinator/s: Dr. Geetha J

Course Contents:

Unit I

Logics and Proofs: The laws of Logic, Logical implication, Rules of inference, Quantifiers, Proofs of theorems.

Unit II

Relations: Relations, Properties of relations, Computer Recognition- Zero-one Matrices and directed Graphs, Equivalence Relations and partitions. POSETS, Hasse Diagrams, Lattices.

Unit III

Combinatorics: Fundamentals of counting, permutation, combination, Combination with repetition, Binomial Coefficient, Principle of inclusion and exclusion, Pigeon hole principle. The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials.

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.

Unit V

Graphs & Trees: Network flow problems, flows and source/sink cuts, Ford-Fulkerson algorithm, Max-flow min-cut theorem, Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes. Applications of Graphs and Trees.

Textbooks:

1. Ralph P. Grimaldi: Discrete and Combinatorial mathematics, 5th Edition, PHI/ Pearson Education, 2004.
2. Kenneth H. Rosen: Discrete Mathematics and its Applications, 7^{th} Edition, McGraw Hill publications.
3. Douglas B. West, Introduction to Graph Theory, Second Edition, Prentice-Hall.

Reference Book:

1. Thomas Koshy: Discrete Mathematics with Applications.

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**
Prerequisites: Basic Mathematics **Contact Hours: 42+28**
Course Coordinator/s: Dr. 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, Undecidability: A language that is not recursively enumerable, an undecidable problem that is RE, definition of Post's Correspondence problem.

Textbook:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Education, 2011.

Reference Books:

1. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.
2. Michael Sipser: Introduction to the Theory of Computation, 3rd Edition, Thompson Course Technology, Boston, MA and Cengage Learning India Pvt. Ltd., 2014.

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 Undecidability, RE languages and Post Correspondence problem. (PO-1,2, *PSO-2)*

Analog & Digital Design

Course Code: CS35 **Credits: 3:0:1**
Prerequisites: Basic Electronics **Contact Hours: 42+28**
Course Coordinator/s: Veena G.S & Aparna R

Course Contents:

Unit I

Op-amps: inside of the op-amp, ideal op-amp versus practical op-amp, performance parameters, Op-amp Application circuits: Inverting amplifier, non-inverting amplifier, voltage follower, summing amplifier, Relaxation Oscillator, R-2R ladder network circuits, binary weighted circuits and Schmitt trigger circuits. Synthesis of logic circuits logic circuits with SOP and POS, K-map, strategy of minimization, minimization of SOP, POS forms, in-completely specified functions.

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.

Textbooks:

1. Stephen Brown, ZvonkoVranesic: Fundamentals of Digital Logic Design with VHDL, Tata McGraw Hill, 3rd Edition, 2012.
2. Anant Agarwal JefferyLang: Foundations of Analog and Digital Electronic Circuits 2005 by Elsevier Inc.15.
3. Anil K Maini, Varsha Agarwal: Electronic Devices and Circuits, Wiley, First Edition, 2009.

Reference Books:

1. Robert L Boylestad, Louis Nashelsky: Electronic devices and circuit theory, 9th edition. 2007.
2. Albert Malvino& David J Bates: Electronic Principles, TMH, 7th edition, 2007.
3. David A Bell: Electronic devices and Circuits, PHI,4th edition, 2006.

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

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)

Object Oriented Programming

Course Code: CS36 **Credits: 3:0:0**
Prerequisites: C Programming **Contact Hours: 42**
Course Coordinator/s: Hanumantharaju R

Course Contents:

Unit -1

The Object Model Foundations of the Object Model: Object-Oriented Programming, Object-Oriented Design, Object-Oriented Analysis Elements **of the Object Model:** Abstraction, Encapsulation, Modularity, Hierarchy, Typing, Concurrency, Persistence Applying the Object Model. Introduction to Java Programming: Java Buzzwords, Overview of Java Datatypes, Variables, arrays, Control statements.

Unit -2

Java Programming Fundamentals: Object-Oriented Programming, The Three OOP Principles, Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Introducing Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes.

Unit -3

Inheritance, Packages & Interfaces
Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Packages, Access Protection, Importing Packages, Interfaces, String and StringBuffer Handling.

Unit -4

Exception Handling: Exception-Handling Fundamentals, Exception Classes, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch clauses, Nested try Statements, throw, throws, finally. **Multithreaded Programming:** Java Thread Classes, The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Suspending, Resuming and Stopping Threads.

Unit -5

Event Handling, Introducing Swing:
Two Event Handling Mechanisms, The Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Event Model, Adapter Classes, Inner Classes. Swing: Introducing Swing. **Lambda Expressions:** Fundamentals, Block Lambda expressions, Passing Lambda Expressions as Argument, Lambda Expressions and Exceptions, Method References.

Textbooks:

1. Object-Oriented Analysis And Design With applications, Grady Booch, Robert A Maksimchuk, Michael W Eagle, Bobbi J Young, 3rd Edition, 2013, Pearson education, ISBN :978-81-317-2287-93.

2. The Complete Reference - Java, Herbert Schildt 10th Edition, 2017, TMH Publications, ISBN: 9789387432291.

Reference Book:

1. Head First Java, Kathy Sierra and Bert Bates, 2nd Edition, 2014, Oreilly Publication , ISBN : 978817366602

Course Outcomes (COs):

This course uses assigned readings, lectures, and homework to enable the students to:

1. Examine the fundamental elements of object model and identify classes and objects for object-oriented programming. (PO-2,3,5 PSO-3)
2. Explore the OOP principles and basic constructs of Java language. (PO-2,3,5 PSO-3)
3. Develop java programs using inheritance, interfaces, and packages. (PO-2,3,5 PSO-3)
4. Explore the exception handling mechanism and thread synchronization. (PO-2,3,5 PSO-3)
5. Design the GUI application using swings and handle the interactions. (PO-2,3,5 PSO-3)

Object Oriented Programming Laboratory

Course Code: CSL37 **Credits: 0:0:1**
Prerequisites: Nil **Contact Hours: 28**
Course Coordinator/s: Dr. Geetha J

Prerequisites: C Programming

Course Contents:

1. Primitive Data type and Operators
2. Control Statements and Arrays
3. Classes, objects, static variables
4. Overloading, overriding, dynamic method dispatch
5. Inheritance, interface
6. Packages & abstract classes
7. String handling
8. Exception Handling
9. Multi-threading
10. Synchronizing the threads
11. Event Handling
12. Programs on Swings
13. Lamda Expressions
14. Collection Frameworks

Course Outcomes (COs):
This course uses assigned readings, lectures, and homework to enable the students to:

1. Develop Java applications using OOP principles. (PO-2,3,5 PSO-3)
2. Design real world applications using multi-threading and exception handling mechanism. (PO-2,3,5 PSO-3)
3. Develop interactive GUI applications using Swings. (PO-2,3,5 PSO-3)

Data Structures Laboratory

Course Code: CSL38 **Credits: 0:0:1**
Prerequisites: Nil **Contact Hours: 28**
Course Coordinator/s: Dr. Parkavi A

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

Textbook:

1. Horowitz, Sahni, Anderson-Freed: Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2008.

Reference Book:
1. Yedidyah, Augenstein, Tannenbaum: Data Structures Using C and C++, 2nd Edition, Pearson Education, 2003.

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:** 3:1:0
Prerequisite: Calculus & Probability **Contact Hours:** 42+28
Course Coordinators: Dr. N. L. Ramesh & Dr. A. Sreevallabha Reddy

Course Contents:

Unit I

Finite Differences and Interpolation: Forward, Backward differences, Interpolation, Newton-Gregory Forward and Backward Interpolation formulae, Lagrange's interpolation formula and Newton's 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's 1/3rd rule, Simpson's 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, Uniform distribution, Exponential distribution, Gamma distribution and Normal distribution.

Unit III

Joint probability distribution: Joint probability distribution (both discrete and continuous), Conditional probability, 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 and Statistical Inference : 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.

Textbooks:

1. R.E. Walpole, R. H. Myers, R. S. L. Myers and K. Ye – Probability and Statistics for Engineers and Scientists – Pearson Education – Delhi – 9th edition – 2012.
2. B.S.Grewal - Higher Engineering Mathematics - Khanna Publishers – 44th edition-2017.

Reference Books:

1. Erwin Kreyszig - Advanced Engineering Mathematics-Wiley-India publishers- 10th edition-2015.
2. Sheldon M. Ross – Probability models for Computer Science – Academic Press – 2009.
3. Murray R Spiegel, John Schiller & R. Alu Srinivasan – Probability and Statistics – Schaum's outlines -4nd edition-2013.
4. Kishor S. Trivedi – Probability & Statistics with Reliability, Queuing and Computer Science Applications – John Wiley & Sons – 2nd edition – 2008.

Course Outcomes (COs):

At the end of the Course, students will be able to

1. Find functional values, derivatives, areas and volumes numerically from a given data. (PO-1,3 PSO-2)
2. Analyze the given random data and their probability distributions. (PO-1,3 PSO-2)
3. Calculate the marginal and conditional distributions of bivariate random variables and determine the parameters of stationary random processes. (PO-1,3 PSO-2)
4. Use Markov chain in prediction of future events and in queuing models.(PO-1,3 PSO-2)
5. Use Sampling theory to make decision about the hypothesis. (PO-1,3 PSO-2)

Design and Analysis of Algorithms

Course Code: CS42 **Credits: 3:1:0**
Prerequisites: Data Structures & Programming Language **Contact Hours: 42+28**
Course Coordinator/s: Pramod C Sunagar

Course Contents:

Unit I

Asymptotic Bounds and Representation problems of Algorithms: Computational Tractability: Some Initial Attempts at Defining Efficiency, Worst-Case Running Times and Brute-Force Search, Polynomial Time as a Definition of Efficiency, Asymptotic Order of Growth: Properties of Asymptotic Growth Rates, Asymptotic Bounds for Some Common Functions, A Survey of Common Running Times: Linear Time, O(n log n) Time, Quadratic Time, Cubic Time, O(nk) Time, Beyond Polynomial Time, Sub linear Time. Some Representative Problems, **A First Problem: Stable Matching:** The Problem, Designing the Algorithm, Analyzing the Algorithm, Extensions, Implementing the Stable Matching Algorithm, Using Lists and Arrays: Arrays and Lists, Five Representative Problems: Interval Scheduling, Weighted Interval Scheduling, Bipartite Matching, Independent Set, Competitive Facility Location.

Unit II

Graphs & Divide and Conquer: Graph Connectivity and Graph Traversal, Breadth-First Search: Exploring a Connected Component, Depth-First Search, Implementing Graph Traversal Using Queues and Stacks: Implementing Breadth-First Search, Implementing Depth-First Search, An Application of Breadth-First Search: The Problem, Designing the Algorithm, Directed Acyclic Graphs and Topological Ordering: The Problem, Designing and Analyzing the Algorithm, A First Recurrence: The Merge sort Algorithm: Unrolling the Merge sort Recurrence, Counting Inversions: The Problem, Designing and Analyzing the Algorithm.

Unit III

Greedy Algorithms: Interval Scheduling: The Greedy Algorithm Stays Ahead: Designing a Greedy Algorithm, Analyzing the Algorithm, Scheduling to Minimize Lateness: An Exchange Argument: The Problem, Designing the Algorithm, Optimal Caching: A More Complex Exchange Argument: The Problem, Designing and Analyzing the Algorithm, Extensions: Caching under Real Operating Conditions, Shortest Paths in a Graph: The Problem, Designing the Algorithm, Analyzing the Algorithm, The Minimum Spanning Tree Problem: The Problem, Designing Algorithms, Analyzing the Algorithms, Huffman Codes and Data Compression: The Problem, Designing the Algorithm.

Unit IV

Dynamic Programming: Weighted Interval Scheduling: A Recursive Procedure: Designing a Recursive Algorithm, Subset Sums and Knapsacks: Adding a Variable: The Problem, Designing the Algorithm, Shortest Paths in a Graph: The Problem, Designing the Algorithm, The Maximum-Flow Problem and the Ford-Fulkerson Algorithm: The problem, Designing the Algorithm, Survey Design: The problem, Designing the Algorithm, Analyzing the Algorithm, Airline Scheduling: The problem, Designing the Algorithm, Analyzing the Algorithm.

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.

Textbook:

1. Algorithm Design - Jon Kleinberg and Eva Tardos, Pearson, 1st Edition (2013).

Reference Book:

1. Introduction to the Design & Analysis of Algorithms - Anany Levitin, 2nd Edition,Pearson Education, 2007.

Course Outcomes (COs):

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

1. Define the basic concepts and analyse 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 analyse 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: 3:0:1**
Prerequisites: Nil **Contact Hours: 42+28**
Course Coordinator/s: Dr. Mohan Kumar

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, Debug Features

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, Stack memory operations

Unit III

Instruction Set, Instruction Usage Examples, implementation of various structures like loop, switch, functions, and subroutines.

Unit IV

Memory System, Exceptions and Interrupts, Interrupt Control and System Control, Application programming, Sensors, Thermistor, LDRs, LEDs 7 segment, LCD, Stepper motor, relays.

Unit V

General Information about the ESP8266: Multitude of boards, Technical specification and SoC datasheets, Scarcity of runtime resources, Boot Process, Known Issues-Real-time clock. Introduction to Micropython language: The Micropython interactive Interpreter mode(aka REPL)Auto-intent, Auto-Completion, Interrupting a running Program, paste mode, Soft reset.

Textbooks:

1. Steve Furber, ARM System-On-Chip Architecture, 2nd edition, Addison Wesley.
2. Wayne Wolf "Computers as Components Principles of Embedded Computer System Design", Second Edition, Elsevier, 2008.
3. Joseph Yiu, "The Definitive Guide to the ARM Cortex-M0 ", 1st edition, Newnes - an imprint of Elsevier,2011.
4. Lyla B. Das, "Embedded systems An Integrated Approach", 2012.

Reference Links:

1. https://docs.micropython.org/en/latest/esp32/general.html,
2. https://docs.micropython.org/en/latest/reference/index.html

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 and Networking

Course Code: CS44 **Credits: 4:0:0**
Prerequisites: Nil **Contact Hours: 56**

Course Coordinator/s: Sanjeetha R

Course Contents:

Unit 1

Data Communications, Networks, Network Types, Network Models – Protocol layering, TCP/IP Protocol Suite, The OSI Model, Physical layer: Date Rate Limits, Performance
Digital to Digital Conversion- Line coding: polar, unipolar, Block coding: 4B/5B, Analog to Digital Conversion -Pulse Code Modulation, Digital to Analog conversion- ASK, FSK,PSK, Analog to Analog conversion – AM,FM,PM. Multiplexing – FDM, TDM, Spread spectrum- FHSS, DHSS
Switching – Circuit switching, packet switching

Unit 2

Error Detection and Correction- Block Coding, Cyclic Codes – CRC, Polynomials, Cyclic code encoder using Polynomials. Checksum, Forward Error Correction: chunk interleaving.
Data Link Layer: Data Link Control –DLC services: framing, Data link layer protocols – stop and wait protocol, Go-Back-N protocol, Selective Repeat Protocol. HDLC , Point to Point Protocol.
Media Access Control – Random Access- CSMA/CD, CSMA/CA, Controlled access, Channelization,

Unit 3

Wired LAN - Standard Ethernet, Wireless LANs- IEEE 802.11- Architecture, MAC sublayer, Addressing mechanism, Connecting Devices
Network Layer: IPV4 Addresses: Address space, Classful Addressing, Classless Addressing, Dynamic Host Configuration Protocol (DHCP), Network Address Translation (NAT).
Routing Algorithms: Link-State (LS) Routing Algorithm, The Distance-Vector (DV) Routing Algorithm

Unit 4

Routing in the Internet: Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter-AS Routing in the Internet: BGP.
Transport layer: Multiplexing and Demultiplexing, Connectionless Transport-UDP: UDP Segment Structure, UDP Checksum, Connection-Oriented Transport-TCP: The TCP Connection, TCP Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management, TCP congestion control.

Unit 5

Application Layer: The Web and HTTP: Overview of HTTP, Non-Persistent and Persistent Connections, HTTP Message Format, User-Server Interaction-Cookies, Web Caching, The Conditional GET. File Transfer- FTP: FTP Commands and Replies, Electronic Mail in the Internet: SMTP, Comparison with HTTP, Mail Access Protocols. DNS—The Internet's Directory Service: Services Provided by DNS, Overview of How DNS Works, DNS Records and Messages, Peer-to Peer Applications: P2P File Distribution

Text Books:

1. Data Communication and Networking, Behrouz A.Forouzan, McGraw Hill, 5 th Edition, 2008.
2. James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach, 6th edition, Addison-Wesley, 2013.
3. Forouzan: Data Communications and Networking, 5th edition, McGraw Hill Education 2013.

Reference Books:

1. Data and Computer Communication, William Stallings, 8th Edition, Pearson Education, 2007.
2. Introduction to Data Communications and Networking – Wayne Tomasi, Pearson Education, 2005.
3. Larry L. Peterson and Bruce S Davie: Computer Networks: A Systems Approach, Fifth Edition, Elsevier, 2011.
4. Tanenbaum: Computer Networks, 4th Ed, Pearson Education/PHI, 2003.
5. William Stallings: Data and Computer Communications, 8th Edition, Pearson Education, 2012.

Course Outcomes (CO'S):

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

1. Identify the different types of network topologies and protocol models. (PO1,2,3,4,PSO-1).
2. Differentiate between digital and analog transmissions. (PO-1,2,3,4, PSO-1).
3. Solve problems in error detection and correction at data link layer. (PO-1,2,3,4,PSO1).
4. Discriminate between different access control methods to shared transmission media. (PO-1,2,3,4, PSO-1).
5. Compare the working of wired and wireless networks. (PO-1,2,3,4, PSO-1).

Computer Organization and Architecture

Course Code: CS45 **Credits: 3:1:0**
Prerequisites: Basics of Computers **Contact Hours: 42+28**
Course Coordinator/s: Chandrika Prasad

Course Contents:

Unit 1

Functional units, bus structures, performance, **Overflow in integer arithmetic**: Numbers, Arithmetic operations and characters, Memory locations and addresses, Memory operations, instructions and **instruction sequencing**, addressing modes, Subroutines and use of stack frames, Encoding of machine instructions.

Unit 2

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. **The Processor:** Processing unit: Fundamental concepts, Execution of complete instruction, Multiple bus organization, Hardwired control, Microprogrammed control.

Unit 3

Instruction level parallelism: Introduction and challenges, **Data dependences and Hazards:** Data dependences, Name dependences, Control Dependences, Basic pipeline scheduling and loop unrolling, Pipelining: Introduction, A simple implementation of a RISC instruction set, The classic five-stage pipeline for a RISC processor, Basic performance issues in pipelining, **The major hurdle of pipelining-** pipeline hazards, Performance of pipeline stages, structural hazards, Data hazards, Branch hazards.

Unit 4

Memory unit: Memory Hierarchy, Basics of Cache memory, Cache mapping techniques, **Multiprocessor Architecture:** Centralized shared-memory architecture, **Distributed memory architecture**, **Basic scheme for enforcing coherence**: Snooping coherence protocols, Basic implementation techniques, An example protocol, Directory based cache coherence protocols, An example Directory protocol.

Unit 5

Input Output Unit: Accessing I/O devices, **Interrupts**: Interrupt hardware, Enabling and disabling of interrupts, Handling multiple devices, Controlling device requests, exceptions, **Direct Memory Access** –Bus arbitration, Synchronous bus and asynchronous bus, parallel port and serial port.

Textbooks:

1. C Hamacher, Z Vranesic, S Zaky: Computer Organization, Tata McGraw Hill, 5th Edition, 2011.
2. John L Hennessy, David A Patterson: Computer Architecture A Quantitative Approach, Elsevier, 5th Edition 2012.

Reference Books:

1. David A. Patterson, John L. Hennessy: Computer Organization and Design, M.K Publishers, 4th edition, 2010
2. William Stallings, "Computer Organization and Architecture, Designing for Performance", 9e, Pearson, 2014.

List of Tutorial 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

Course Outcomes (COs):

1. Describe an overview of computer hardware and software which includes the basic functional units, interconnection, addressing techniques and instruction sequencing. (PO-1,3,5,10 PSO-1,2)
2. Appraise different algorithms used to perform fast multiplication, division and to represent floating point numbers in binary. (PO-1,2,3,4,5,10 PSO-1,2)
3. Examine basic processing unit and importance of pipelining to achieve instruction level parallelism. (PO-1,2,3,4,5,10 PSO-1,2)
4. Illustrate cache memory mapping techniques, various memory architectures and protocols for cache coherence. (PO-1,2,3,4,5,10 PSO-1,2)
5. Explain the basics concepts of I/O, interrupts, direct memory access technique and types of busses. (PO-1,3,5,10 PSO-1,2)

Software Engineering

Course Code: CS46 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Dr. Shilpa Chaudhari

Course Contents:

Unit 1

Software Process: The Changing Nature of Software - WebApps, Mobile Applications, Cloud Computing, Product Line Software; Software Process - The Process Framework, Umbrella Activities, Process Adaptation.
Process Models: Prescriptive Process Models - The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models; Specialized Process Models - Component-Based Development, The Formal Methods Model, Aspect-Oriented Software Development; The Unified Process;
Agile Development - What Is Agility? Agility and the Cost of Change, What Is an Agile Process?, Extreme Programming, Other Agile Process Models – Scrum, Dynamic Systems Development Method, Agile Modeling, Agile Unified Process; A Tool Set for the Agile Process;
SPI Process, CMMI; Case study on current problem statement of software development.

Unit 2

Understanding Requirements: Requirements Engineering, Eliciting Requirements, Developing Use Cases, Building the Analysis Model, Negotiating Requirements, Requirements Monitoring, Validating Requirements, Avoiding Common Mistakes.
Scenario-Based Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case.
Requirements Modeling for Web and Mobile Apps, Applying requirement engineering on the same case study of Unit-1.

Unit 3

Design Concepts: The Design Process, Design Concepts, Design Model.
Architectural Design: Software Architecture, Architectural Genres, Architectural Styles, Architectural Considerations, Architectural Decisions, Architectural Design;
User Interface: The Golden Rules of User Interface Analysis and Design, WebApp and Mobile Interface Design; Applying design modeling by taking requirement specification from Unit-2

Unit 4

Quality Concepts: Software Quality, Software Quality Dilemma, Achieving Software Quality, Formal Technical Reviews
Software Project Estimation - Observations on Estimation, The Project Planning Process Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Estimation for Object-Oriented Projects, Estimation for Object-Oriented Projects; Project Scheduling –Scheduling.
Risk Management: Reactive versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management.
Computation of relevant metrics for the case study on current problem statement of software development considered in Unit-1.

Software Maintenance, Software Supportability, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering.

Unit 5

Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Test Strategies for MobileApp, Validation Testing, System Testing,
Testing Conventional Applications: Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing, Model-Based Testing, Testing Documentation and Help Facilities
Testing OOA and OOD Models, Object-Oriented Testing Strategies, Object-Oriented Testing Methods; Designing test cases as per the requirement specification from Unit-2

Textbook:

1. Roger S Pressman and Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 8/e, New York: McGraw-Hill, 2015

Reference Books:

1. Ian Sommerville, Software Engineering, 10th Edition,Pearson, 2016
2. Emilia Mendes, Nile Mosley: Web Engineering, Springer, 2006
3. David Gustafson: Software Engineering, Schaum's Outline Series, McGraw Hill, 2002

Course Outcomes (COs):

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

1. Explain the various aspects of the software process, considering the important process models and addressing the debate between prescriptive and agile process philosophies. (PO-1,4,6,7,8,9,11 PSO-2,3)
2. Understand the analysis and design methods using object-oriented techniques and UML modeling, focusing on web/mobile applications. (PO-1,2,3,4,6,7,8,9,10,11 PSO-2,3)
3. Decide all aspects of quality assurance, formal verification techniques, and software maintenance. (PO-1,4,5,6,7,8,9,10,11 PSO-2,3)
4. Compare the testing strategies required at each phase of software development for a particular case study. (PO-2,3,4,5,6,9,10,11 PSO-2,3)
5. Apply the concepts of software engineering principles on the real-world problems of software development using appropriate tools. (PO-1,2,3,4,5,6,7,8,9,10,11 PSO-2,3)

Algorithms Laboratory

Course Code: CSL47 **Credits: 0:0:1**
Prerequisites: Data Structures, C++, C **Contact Hours: 28**

Course Coordinator/s: Pramod Sunagar & Srinidhi H

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.
4. Brute Force techniques.
5. Graph traversal techniques.
6. Divide and Conquer Techniques.
7. Greedy Algorithms.
8. Dynamic Programming.
9. Branch and Bound Techniques.

Reference Books:

1. Algorithm Design - Jon Kleinberg and Eva Tardos, Tsinghua University Press (2005).
2. Anany Levitin: Introduction to the Design & Analysis of Algorithms, 2nd Edition, Pearson Education, 2007.

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)

Python Programming Laboratory

Course Code: CSL48 **Credits: 0:0:1**
Prerequisites: Nil **Contact Hours: 28**
Course Coordinator/s: Hanumantharaju R & Pradeep Kumar D

Course Contents:

There shall be a minimum of 2 exercises conducted on each of the following topics.

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. GUI/Network Programming
12. Python for Data Science
13. Numpy and Pandas
14. Data wrangling Application

Reference Books:

1. Mark Lutz: Learning Python, 5th Edition, Orielly Publications 2013.
2. John Zelle: Python Programming: An Introduction to Computer Science, 2nd Edition.
3. Paul Barry, Head First Python, O‟Reilly Publication, 2010.
4. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinny, O'Reilly Media, 2012. ISBN 978-1-4493-1979-3

Course outcomes:

1. Use internal and external Python libraries, data structures, functions inherent to Python in-order to handle data. (PO-1, 2, 3,5) (PSO-2, 3)
2. Develop real-world applications using oops, files and exception handling provided by python (PO-1, 2, 3, 5) (PSO-2, 3)
3. Apply Python as a scripting language to analyze huge datasets, apply data science related statistics on datasets (PO-1, 2, 3,5) (PSO-2, 3)