# CURRICULUM
# FOR THE ACADEMIC YEAR 2020-2021

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## B.E. III SEMESTER

## POOJYA DODDAPPA APPA COLLEGE OF ENGINEERING
## (An autonomous college under VTU)
## KALABURAGI

## SCHEME OF TEACHING FOR III SEMESTER -2020-2021

### B.E. (COMPUTER SCIENCE AND ENGINEERING)

| Code No. | Course | Hours/Week | | | | Maximum Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | Lecture | Tutorial | Practical | Credit | CIE | SEE | Total |
| SEMESTER III | | | | | | | | |
| 19MA31 | Computational Mathematics - I | 2 | 2 | 0 | 3 | 50 | 50 | 100 |
| 19CS32 | Mathematical Foundations of Computer Science | 3 | -- | 0 | 3 | 50 | 50 | 100 |
| 19CS33 | Object Oriented Programming | 4 | -- | 0 | 4 | 50 | 50 | 100 |
| 19CS34 | Data Structures | 4 | 0 | 0 | 4 | 50 | 50 | 100 |
| 19CS35 | Analog and Digital Electronics | 3 | 2 | 0 | 4 | 50 | 50 | 100 |
| 19HU36 | Constitution of India and Professional Ethics | 2 | -- | -- | 0 | 50 | 0 | 50 |
| 19KAN37 | Kannada | 1 | -- | -- | 1 | 50 | 0 | 100 |
| 19CSL31 | Object Oriented Programming Lab | 0 | 0 | 3 | 1 | 50 | 50 | 100 |
| 19CSL32 | Data Structures Lab | 0 | 0 | 3 | 1 | 50 | 50 | 100 |
| 19CSL33 | Analog and Digital Electronics Lab | 0 | 0 | 3 | 1 | 50 | 50 | 100 |
| | **Total** | **19** | **6** | **9** | **22** | **500** | **400** | **900** |

**AUTONOMOUS SYLLABUS FOR B.E III SEMESTER 2020-2021**

| **Course Title: COMPUTATIONAL MATHEMATICS - I** | | |
|---|---|---|
| Subject Code : **19MA31** | Credit : 03 | CIE: 50 |
| Number of Lecture Hours/Week | **2L+1T** | SEE: 50 |
| Total Number of Lecture Hours | **28** | SEE Hours: 03 |

**Prerequisites:** Students should have knowledge of Differential calculus, Integral calculus and Differential equations.

**Course Objectives:** To enable the students to obtain the knowledge of Engineering Mathematics in the following topics

- Interpolation methods , Numerical differentiation and Numerical integration
- Fourier Series and Z-transformation and its application in engineering fields
- Methods of least squares to fit straight line and second degree parabola
- Solve the problems using probability theory

| **MODULES** | **Teaching Hours** |
|---|---|
| **Module I**<br>**Finite differences: (**Forward and Backward differences), Interpolation, Newton's Forward and Backward formulae. Lagrange's interpolation and inverse interpolation formulae.<br>**Numerical differentiation:** Numerical differentiation using Newton's forward and backward interpolation formulae and problems.<br>**Numerical integration**: Trapezoidal rule, Simpsons $1/3^{rd}$ and $3/8^{th}$ rule, Weddle's rule (all formulae and rules without proof) | **6 hours** |
| **Module II**<br>**Difference equations and Z-Transforms :**<br>Difference equations –Basic definitions, Z-Transform-Definitions, standard Z-transform, linearity property , damping rule, shifting rule , initial value theorem ,final value theorem . Inverse Z-Transform and problems. | **6 hours** |
| **Module III**<br>**Fourier series:** Periodic functions, Fourier series with periods $(0, 2\pi)$, $(-\pi, \pi)$, $(0, 2l)$ and $(-l, l)$. Half range Fourier series, Practical harmonic analysis and problems. | **6 hours** |
| **Module IV**<br>**Optimization techniques:** Linear Programming, Mathematical formulation of linear programming problem(LPP), Types of solutions, Graphical Method, basic feasible solution, canonical and standard forms and simplex method. | **5 hours** |
| **Module V**<br>**Time Series and Forecasting:** Moving averages, smoothening of curves, | |

| | |
|---|---|
| forecasting models and methods, Statistical Quality Controls methods.<br>**Testing of Hypothesis:** Test of significance, Chi-Square test, T-Test, ANOVA,F-Test, applications to medicine, agriculture, etc.. | **5 hours** |

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module**.**

**TEXT BOOKS:**

Higher Engineering Mathematics by B.S.Grewal, Khanna publishers; 40th Edition.2007**.**

**REFERENCES:** 1.Advanced Engineering Mathematics by E. Kreyszig, John Willey & sons 8th Edn.
2.A short course in differential equations – Rainvile E.D.9th Edition.
3.Advanced Engineering Mathematics by R.K.Jain & S.R.K Iyengar; Narosa publishing House.
4.Introductory methods of numerical analysis by S.S.Sastry
5. Probability statistics and Queuing theory- Kishore Trivedi./??check the title

**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| **19MA31** | **CO1** | Compute derivatives of the functions numerically using given data Computation of interpolation polynomials and numerical integration. |
| | **CO2** | Analyze discrete type system using convolution and the Z-transform. |
| | **CO3** | Construction of Fourier series for periodic signals and Fourier series to analyze circuits. |
| | **CO4** | Apply optimization techniques for real life problems |
| | **CO5** | Apply Statistical control methods and apply LPP for real-life problems in agriculture, medicine etc. |

| **Course Title: MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE** | | |
|---|---|---|
| Subject Code : **19CS32** | Credit : 03 | CIE: 50 |
| Number of Lecture Hours/Week | 03 | SEE: 50 |
| Total Number of Lecture Hours | 42 | SEE Hours: 03 |
| **Prerequisites:** Engineering Mathematics | | |
| **Course Objectives:**<br>To develop mathematical thinking and problem-solving skills associated with writing proofs.<br>To expose students to a wide variety of mathematical concepts those are used in the Computer Science discipline. | | |
| **MODULES** | | **Teaching Hours** |
| **Module I**<br><br>**Mathematical logic**: Basic Connectives and truth tables, Logic Equivalence-The Laws of logic, Logical Implications-Rules of Inference.<br>**Counting:** Permutations, combination, Pigeonhole, Principles.<br>**Relations and Digraphs**: Product Sets and Partitions, Relation and Digraphs, Properties of Relations and Digraphs Properties of Relations, Equivalence Relations, Data structures for Relations and Digraphs, Operations on Relations, Transitive Closure and Warshall"s Algorithm. | | **09 hrs** |
| **Module II**<br><br>**Function:** Function, Function for Computer Science, Growth of functions, Permutation Functions<br>**Order Relations and Structure:** Partially Ordered Sets, External Elements of Partially, Ordered Sets, Lattices, Finite Boolean Algebras, Functions on Boolean Algebras, Circuit. | | **08 hrs** |
| **Module III**<br><br>**Introduction to Graph Theory-I:** Definition & Examples, Sub-graph, complements and graph Isomorphism, Vertex degree, Euler trails and circuits.<br>**Graph Theory-II:** Planar graphs, Hamilton paths and cycles, Graph coloring, chromatic polynomials, Transport networks. (Problem solving using C) | | **08 hrs** |
| **Module IV**<br><br>**Trees:** Definitions, Properties, and Examples Rooted Trees, pre order traversals and post order traversals, Trees and Sorting, Weighted Trees and Prefix Codes, minimal spanning tree.<br>**Languages and finite state machines:** Languages, representations of special grammars and languages, finite state machine, semi groups machines and languages | | **08 hrs** |
| **Module V**<br><br>**Algebraic structures:** Semigroups, monoids, definition, example and elementary properties, Homomorphism, isomorphism and cyclic groups, cosets and lagranges theorem, elements of coding theory, the hamming matric, parity check and | | **09 hrs** |

| | |
|---|---|
| generator matrices, **Groups coding:** coding with coset headers and hamming matrices. Decoding in cosets: the cycle index, polays method of enumeration. | |

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module**.**

**TEXT BOOKS:**

1. Grimaldi R. P., *"Discrete and Combinatorial Mathematics"*, 6th edition, Pearson Education 2004.
2. B.Kolman and R.C.Busby, "*Discrete Mathematical Structures for Computer Science*", PHI, New Delhi, 1994.

**REFERENCES:**

1. Frank Harary, *"Graph Theory"*, Addison Wesley Publishing Company, 1995.
2. C. L. Liu C. L., *"Elements of Discrete Mathematics"*, 2nd edition , McGraw Hill, Singapore, 1985.
3. J.P. Tremblay, *"Discrete Mathematical Structures with Applications to Computer Science"*, McGraw Hill, N.Y., 1977
4. Kenneth H Rosen, *"Discrete Mathematics and its applications",* 6th Edition, McGraw Hill 2007.

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| **19CS32** | **CO1** | Acquire knowledge of mathematical logic, proofs of basic discrete probability, number theory and apply in problem solving |
| | **CO2** | Apply various concept of functions and relations for solving computing problems |
| | **CO3** | Demonstrate knowledge of fundamental concept in graphs , trees and its properties using various modeling techniques |
| | **CO4** | Design grammars, finite state machines and its algebraic structures. |
| | **CO5** | Discuss recurrence relations , generating functions, algebraic systems and their applications in coding theory and group theory |

| Course Title: OBJECT ORIENTED PROGRAMMING | | |
|---|---|---|
| **Subject Code : 19CS33** | Credit :4 | CIE: 50 |
| Number of Lecture Hours/Week | 4 | SEE: 50 |
| Total Number of Lecture Hours | 52 | SEE Hours: 03 |
| **Prerequisites: Programming in C** | | |

**Course Objectives:**

- Formulate the problems in a better way giving high reliability, adaptability and extensibility to the applications.
- Provides a base for the further object oriented programming like Java, C# and .Net.

| MODULES | Teaching Hours |
|---|---|
| **Module I**<br>**Object-Oriented Paradigm**: Introduction to program paradigms, OOP's , A new paradigm, Evolution of programming paradigms, Structured versus object-oriented development, Elements of object-Oriented programming, Objects, Classes, Multiple views of the same object, Encapsulation and data abstraction, Inheritance, Delegation – Object composition, Polymorphism**.**<br>**Arrays and Strings:** Introduction, Operations on arrays, Array illustrations, Multi-dimensional arrays, Strings, Strings manipulations, Arrays of strings, Evaluation order/Undefined behaviors. | **11 Hrs** |
| **Module II**<br>**Modular programming with functions**: Introduction, Function components, Passing data to functions, Function return data type, Library Functions, Parameter passing, Return by reference, Default arguments, Inline functions, Function overloading, Function Templates, Arrays an functions, C++ Stack, Scope and extent of variables, Storage Classes, Functions with variables number of arguments.<br>**Classes and Objects**: Introduction, Class Specification, Class Objects, Accessing class members, Defining member functions, Outside member functions as inline, Accessing member functions within the class, Data hiding, Access boundary of objects revisited, Empty classes, Pointers within a class, Passing objects as arguments, Returning objects from functions, friend functions and friend classes, Constant parameters and member functions, Structures and Classes. | **10 Hrs.** |
| **Module III**<br>**Object Initialization:** Constructors, Parameterized constructors, Destructor, Constructor overloading, Order of construction and destruction, Constructors with default arguments, Dynamic initialization through constructors, Constructors with dynamic operations, Copy constructor, Constructors for two-dimensional arrays, Constant objects and constructor, Static data members with constructors and destructors.<br>**Operator Overloading:** Introduction, Overloadable operators, Unary operator overloading, Operator keyword, Operator return values, Binary operator | **10 Hrs.** |

| | |
|---|---|
| overloading, Arithmetic operators, Concatenation of strings, Comparison operators, Arithmetic assignment operators, Overloading of new and delete operators, Data conversion, Overloading with friend functions, Assignment operator overloading. | |
| **Module IV**<br>**Inheritance and Virtual Functions:**<br>Introduction, Derived class declaration, Forms of inheritance, Inheritance and member accessibility, Constructors in derived classes, Destructors in Derived classes, Abstract Classes, Multilevel inheritance, multiple inheritance, Hierarchical inheritance, Multipath inheritance and virtual base class, Hybrid inheritance.<br>**Virtual Functions**: Introduction, Need for virtual functions, Pointer to derived class objects, definition of virtual functions, Array of pointers to base class objects, pure virtual functions. | **10 Hrs.** |
| **Module V**<br>**Generic Programming with Templates**:<br>Introduction, Function templates, Overloaded Function Templates, Nesting of Function Calls, Multiple Argument Function Template, Class Templates, Inheritance of class templates, Class Templates with Overloaded Operators.<br>**Stream Computations with Console-**<br>Introduction, Predefined Console Streams, Hierarchy of Console Stream Classes, Unformatted Console I/O Operations, Formatted Console I/O Operations., Manipulators, Stream Operators with User-defined Classes<br>**Stream Computation with Files** –Introduction . Hierarchy of File Stream Classes, Opening and Closing of Files, File Modes File Pointers and their Manipulations. | .<br>**11 Hrs.** |

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**TEXT BOOKS:**


1. K.R.Venugopal, Rajkumar, T . Ravishankar , " Mastering C++" , Tata McGraw-Hill Publishing Company Ltd. New Delhi.
2. E. Balagurusamy , "Object-Oriented Programming with C++", 2nd Edition , Tata McGraw-hill Publishing Company Ltd. New Delhi


**REFERENCES:**


1. Herbert Schildt, "The complete reference C++", 4th Edition, Tata McGraw-Hill Publishing Company Ltd. New Delhi, 2005. University Press, 2006.
2. Stanley B Lippman, Josee Lajoie, "C++ Primer" , Barbara E Moo, 4th Edition, Addison Wesley, 2005
3. Object-Oriented Programming with C++, Sourav Sahay, Oxford University Press, 2006.


**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CS33 | CO1 | Illustrate The object oriented programming concepts and Differentiate between OOPS and conventional structured programming approaches. |
| | CO2 | Create structured, modular and re-usable code. |
| | CO3 | Demonstrate Object features and Reusability using operator overloading. |
| | CO4 | Explain Dynamic Object creation using Virtual functions. |
| | CO5 | Adopt generic programming and streams for developing Applications. |

| **Course Title: DATA STRUCTURES** | | |
|---|---|---|
| Subject Code : **19CS34** | Credits :4 | CIE: 50 |
| Number of Lecture Hours/Week | 4 Hrs (Theory) | SEE: 50 |
| Total Number of Lecture Hours | 52 | SEE Hours: 03 |

**Prerequisites:**

- C language fundamentals and programming skill.
- Basic knowledge of algorithm development
- Knowledge of linear and Non-linear data types

**Course Objectives:**

- To understand the behavior of data structures such as stacks, queues, trees, hash tables, search trees and their representations.
- To choose the appropriate data structure for a specified application.
- To analyze various searching and sorting algorithms.

| **MODULES** | **Teaching Hours** |
|---|---|
| **Module - I**<br>**Structures and Unions:** Structure definition, giving value to members, Structure initialization, Comparison of structure variables , Arrays of structures, Arrays within structures, Structure within structures, Structure and functions, Unions, Size of structures, Bit-fields.<br>**Pointers:** Understanding pointers, and the address of operator, Declaring and initializing pointer, Accessing a variable through it's pointer, Pointer and arrays, Pointer and character strings, Pointer and functions, Pointer and Structures.<br>**Dynamic memory allocation:** Meaning of dynamic memory allocation, MALLOC, CALLOC, Free and REALLOC functions, Pointer revisited.<br>**File management:** Definition and opening a file, closing a file, I/O operations on files, Error handling during file operation, Radom access to files, Command line arguments | **10 Hrs** |
| **Module - II**<br>**Definition and Representing Stack in C :** Primitive operation, Example. Implementing the pop() operation, Testing for exceptional conditions, Implementing the push() operation, Example: Infix, Postfix and Prefix, Basic definitions and Examples, Evaluating a postfix expression, Program to evaluate postfix expression , Converting an expression from infix to postfix, Program to convert expression from infix to postfix.<br>**Recursive definition and processes:** Factorial function, Multiplication of natural numbers, Fibonacci sequence, Binary search, Properties of recursive definition or algorithm Recursion in C: Factorial of a number Generation of Fibonacci numbers, Binary searching, Towers of Hanoi problem. | **10 Hrs** |

| | |
|---|---|
| **Module – III**<br>**The queue and it's sequential representation:** C implementation of queues, Insert operation, Priority queues, Array implementation of priority Linked lists: Inserting and removing nodes from a list. Linked implementation of stacks, Get node and Free node operations, Linked list implementation of queues, Linked list as a data structure, Example of list operations, Header nodes. Array implementation of list, Linked implementation of lists. Limitations of array implementation, Allocating and freeing dynamic variables, Linked list using dynamic variable, Queues as lists in C, Example of list operations in C, Non- integer and non -homogeneous lists. . | **10 Hrs** |
| **Module - IV**<br>**Other list structures:** Circular lists, Stack as circular list, Queues as a circular list, Primitive operations on circular list, doubly linked list.<br>**Binary trees:** Operations on binary trees and applications of binary trees Binary tree representation: Node representation of binary tree, Internal and external nodes, Implicit array representation of binary trees, Choosing a binary tree representation, Binary tree traversals in C, Threaded Binary trees.<br>**Trees and their applications:** C representation of trees, Tree traversals, General expression as trees, Evaluating an expression tree, Constructing a tree. | **11 Hrs** |
| **Module - V**<br>**Sorting & Searching:** Binary tree sort, Simple insertion sort, Address calculation sort, Radix sort. Sequential searching, Searching an ordered table, Indexed sequential search, Interpolation search. **Tree searching :** Inserting into a binary search tree, Deleting from a binary search tree.<br>**Hashing:** Resolving hash clashed by open addressing, Choosing a hash function. | **11 Hrs** |

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text book :**


1. E. Balgurusamy , "*Programming in ANSI C*", 7 th Edition, Tata McGraw-Hill Publication, 2017.
2. Yedidyah Langsam, Moshe J. Augenstein and Aaron M. Tannenbaum, "*Data Structures Using C and C++"*, 2nd Edition, Prentice-Hall of India publication, 2005.


**Reference Books:**


1. Debasis Samanta, "Classic Data Structures", 2nd Edition, PHI, 2009.
2. Richard F. Gilberg and Behrouz A. Forouzan:, "Data Structures A Pseudocode Approach with C", Cengage Learning, 2005.
3. Robert Kruse & Bruce Leung, "Data Structures & Program Design in C", Pearson Education, 2007.
4. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition, Pearson Education, 2007.


**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| | **CO1** | Apply the fundamental knowledge of pointers, dynamic memory allocation and recursion for designing data structures. |

| | | |
|---|---|---|
| **19CS34** | **CO2** | Demonstrate the usage of stack, queue data structure for design of applications. |
| | **CO3** | Illustrate basic operations on linked lists and construct various data structures using linked lists. |
| | **CO4** | Design Binary trees and binary search trees using tree data structure. |
| | **CO5** | Compare, analyze and implement different sorting and searching Techniques. |

| **Course Title: Analog and Digital Electronics** | | |
|---|---|---|
| Subject Code : **19CS35** | Credits :4 | CIE: 50 |
| Number of Lecture Hours/Week | 3 Hrs (Theory) +2 (Tut) | SEE: 50 |
| Total Number of Lecture Hours | 42 | SEE Hours: 03 |

**Prerequisites:**

- Knowledge of Basic Electronics and Boolean algebra.

**Course Objectives:**

- Recall and Recognize characteristics of PDs, optocouplers, BJT.
- Demonstrate and analyze operational amplifier circuits and their applications
- Describe and analyze combinational logic circuits, simplifications of algebraic equations using Karnaugh maps and Quine McClaskey techniques
- Design decoders, encoders and substractors , Binary comparators latches and flip flops.
- Design registers and counters , A/D and D/A converter.

| **MODULES** | **Teaching Hours** |
|---|---|
| **Module - I**<br>Photo diodes , Light emitting diodes, Optocouplers, BJT Biasing: Fixed Bias , Collector to Base Bias, Voltage Divider Bias, Operational Amplifiers, Application circuits: Multivibrators using 555 IC, Peak detector, Schmitt trigger, Active filters, Non linear Amplifier, Relaxation Oscillator, Current to voltage, voltage to Current converter, Regulated power supply parameters, Adjustable voltage Regulator. | **10 Hrs** |
| **Module - II**<br>The Basic Gates: Review of Basic Logic gates, Positive and Negative Logic, . Combinational Logic Circuits: Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Hazards and Hazard covers | **8 Hrs** |
| **Module – III**<br>Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays,<br>Arithmetic Building Blocks: Half-adder, Full adder, Adder &Substractor, arithmetic logic unit.<br>Flip- Flops: RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOP,T FLIP-FLOP, Edge-triggered JK FLIP-FLOP. | **8 Hrs** |
| **Module - IV**<br>Flip- Flops: FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs. Registers: Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers. Counters: Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus. | **8 Hrs** |

| **Module - V**<br>Counters: Decade Counters, Presettable Counters, Counter Design as a Synthesis problem, A Digital Clock. D/A Conversion and A/D Conversion: Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, A/D Accuracy and Resolution. | **8 Hrs** |
|---|---|

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module**.**

**Text Books:**

1. Anil K Maini, Varsha Agarwal, "*Electronic Devices and Circuits*", Wiley, 2012.
2. Donald P Leach, Albert Paul Malvino&GoutamSaha, "*Digital Principles and Applications*", 8th Edition, Tata McGraw Hill, 2015

**Reference Books:**

1. R D Sudhaker Samuel, "*Illustrative Approach to Logic Design*", Sanguine-Pearson, 2010.
2. M Morris Man, "*Digital Logic and Computer Design"*, 10th Edition, Pearson, 2008.

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| **Course Code** | **CO #** | **Course Outcome (CO)** |
|---|---|---|
| **19CS35** | **CO1** | Explain LED optocouples, BJT bias , op amp and its application. |
| | **CO2** | Optimize logic equations using K-maps, Quine Mc-Cluskey method, understand don't care conditions |
| | **CO3** | Design combinational circuits using decoders , encoders, comparartor's PAL. |
| | **CO4** | Differentiate different flip flops, counters, registers, explain its applications. |
| | **CO5** | Design of A/D & D/A converters. |

| **Course Title: CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS** | | |
|---|---|---|
| Subject Code : **19HU36** | Credit : **0** | CIE: 50 |
| Number of Lecture Hours/Week | **2 hrs** | SEE: 50 |
| Total Number of Lecture Hours | **28** | SEE Hours: 02 |
| **Pre-requisites:** | | |
| **Course objectives:**<br>To enable the students to obtain the basic knowledge about The Constitution of India and Professional Ethics in the following topics:<br>• Introduction and Fundamental Rights<br>• Directive Principles of the State Policy and the State Executive<br>• The Union Executive<br>• Constitutional Provisions for women, Children & SC/ST 'S , Emergency<br>• Provisions and Election Process<br>• Engineering Ethics | | |
| **Modules** | | **Teaching Hours** |
| **Module-I**<br>**Introduction and Fundamental Rights** : The Constitution of India. Evolution of the Constitution. The Constituent Assembly of India. Sources and Features of the Indian Constitution. Preamble to the Constitution of India. Salient Features of Fundamental Rights and their classification. General exercise of Fundamental Rights and their limitations. RTI (Right to Information Act of 2005 Under Article 19(1)) and The Right of Children to Free and Compulsory Education Act or Right to Education Act (RTE) Under Article 21-A of the Constitution. Article 371(J) of the Constitution applicable to Hyderabad Karnataka Area. | | **06 Hrs** |
| **Module – II**<br>**Directive Principles of the State Policy and The State Executive**: Under Article 36 to 51 of The Constitution and their Relevance. Fundamental Duties Under Article 51A of The Constitution and their Relevance. State Government - The Governor- Appointment, Powers and Functions of the Governor. The Appointment of Chief Minster, his Powers and Functions. The State Council of Ministers and their Functions. The State legislature and The State Council. The High Court of the State, its Powers and Jurisdiction. Appointment and Qualifications of High Court Judges. | | **06 Hrs** |
| **Module – III**<br>**The Union Executive**: Central Government. The President of India, his Election, Powers and Functions. The Vice-President of India, his Election, Powers and Functions. The Supreme Court of India and its Structure. Appointment and Qualification of Supreme Court Judges. Their Powers and Functions. The Structure of Judiciary in India. The Parliament of India. The Prime Minister, his Appointment, Powers and Functions. The Union Council of Ministers their Powers and Responsibilities. Concept of Public Interest Litigation (PIL) **.** | | **06 Hrs** |

| | |
|---|---|
| | |
| **Module – IV**<br><br>**Constitutional Provisions and Emergency Provisions and Election Process** : Constitutional for Women, Children, Backward Classes and Scheduled Caste and Scheduled Tribes under different Article of The Constitution. Different types of Emergencies under Article 352, 356 and 360 of the Constitution of India. The Election Commission of India- its Powers and Functions. The State Election Commission | **05 Hrs** |
| **Module – V**<br><br>**Engineering Ethics:** Its Aims and Scope, Responsibilities of Engineers, Impediments to their Responsibilities, Honesty, Integrity, Reliability, Risk and Safety Measures, Liabilities of Engineers. | **05 Hrs** |

**Question paper pattern:**
Solve five full questions selecting at least one question from each Module.

**Text books:**

1. An introduction to the constitution of India and Profession Ethics.
   By B. R. Venkatesh and Merunandan K. B. Publisher : Idea International Publication Bangalore.
2. The Constitution of India and Professional Ethics.
   By K. R. Phaneesh. Publisher : Sudha Publication Bangalore.
3. Professional Ethics.
   By S. Chand. Publisher : S. Chand & Company Ltd. Ram Nagar, New Delhi - 110055.

**Reference Books:**

1. Constitution of India and Professional Ethics By : M Raja Ram. Publisher : New Age International(P) Limited, New Delhi.
2. The Constitutional law of India By : J.N.Pandhey . Publisher : Central Law agency , Allahabad.

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| **19HU36** | **CO1** | Explain the evolution and features of constitution, fundamental rights and their classification |
| | **CO2** | Describe the directive principles of state policy, fundamental duties and The State Executive |
| | **CO3** | Describe about The Union Executive and concept of Public Interest Litigation |
| | **CO4** | Explain the Constitutional Provisions for women, children, SC/ST'S, Emergency Provisions and Election Process |
| | **CO5** | Identifies the qualities required for an professional engineers to be ethical |

| **Course Title: OBJECT ORIENTED PROGRAMMING LAB** | | |
|---|---|---|
| Subject Code :**19CSL31** | Credit :1 | CIE: 50 |
| Number of Practical Hours | 3 | SEE: 50 |
| Total Number of Lecture./Practical Hours | | SEE Hours: 03 |
| **Prerequisites: Programming in C** | | |
| **Course Objectives:**<br>1.Handson experience of Object Oriented Programming Concepts with C++.<br>2.Preparedness to study independent layer Object Oriented Programming language and apply to variety of real time problem scenarios. | | |
| | | |

**PART – A**

1. Write a C++ program to display youngest and eldest person, age from a given set of N number of person(array).

2. Write a C++ program to find Addition and Subtraction of two M x N matrices.

3. Write a C++ program to concatenate two strings.

4. Write a C++ program to find factorial of a number using recursive function.

5. Write a C ++ program to store numbers into an array and find sum of all the elements of the array using pointers.

**PART – B**

1. Create a simple class STUDENT containing the data members, rollno, name, age & display the contents using user defined functions. Test the program with and without scope resolution operator.
2. Create a class called, EMPLOYEE containing data members, Empno, Empname, Designation, Basic pay, DA, HRA, Insurance, TAX. Develop the member functions for calculating and displaying the Netsalary (Netsalary=(Basicpay + DA+HRA)-(Insurance + TAX).

3. Write a C++ program to create class DATE. Display age of the person by considering date of birth and current date using inline function.

4. Write a C++ program to create a class ACC with data members, accno, balance. Create objects, namely,ACC1 and ACC2.Writeamemberfunction to transfer amount fromACC1toACC2. Display the new balance in the transacted accounts.
5. Create a class FIXED-DEPOSIT with data members, principal- amt, year, and rate of interest. Using constructors, initialize the principal-amt and rate of interest. Find the gross amount after a given period of investment.

6. Write a C++ program to sort N numbers using swap as friend function.

7. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading +&- and display the contents of stack.
   s1=s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on to top of the stack.
   s1=s1-; where s1 is an object of the class STACK and – operator pops the element.

8. Write a C++ program to create a class NAME and implement the following operations. Display the result after every operation by overloading the <<.
   i) NAME firstname = "Herbert"
   ii) NAME lastname = "Schield"
   iii) NAME fullname = firstname + lastname
   (Use copy constructor)

9. Write a C++program to create a class called MATRIX using a two-dimensional array of integers. Implement the following operations by overloading the operator==which checks the compatibilityoftwomatricesm1andm2tobeaddedand subtracted.Performtheadditionandsubtractionbyoverloadingtheoperators+and–respectively.Displaytheresults.
   if(m1==m2)
   {
   m3=m1+m2;
   m4=m1–m2;
   } else display Error message.

10. Write a C++ program to demonstrate the function overloading by overloading the user defined function called ADD with variable type and number of arguments.
    (Eg. ADD (Int,int), ADD(int, float), ADD(init,int, float) etc.)

11. Create three classes, namely, STUDENT ,EXAM and RESULT. The STUDENT class has data members, Rollno, Name and Branch and the class EXAM inherits the STUDENT class with new own data members, marks scored in six subjects. Derive the RESULT class from the EXAM class and it has its own data members, total_marks.

12. Write an inter active program to model this multilevel inheritance relationship.

13. Create a base class RESERVATION. Create the derived classes, namely, ADULT, SENIOR_CITIZEN, CHILD by multiple inheritance. The RESERVATION class has data members, Name_of_passenger, Age,

| | |
|---|---|
| Date_of_journey, Source, Destination, Ticket_charge. Write an inter active program to display the ticket charges depending upon the category of passenger.<br>(Note: Charge for Children =½ofadultticketcharge. Senior citizen=¼ of adult ticket charge.)<br><br>14. Write a C++ program to create a pure virtual function and show its access from the object of a derived class through the pointer of base class.<br><br>15. Write a C++ program to sort a given array of N numbers using template function.<br><br>Note: For SEE, students will be asked to do similar programs from PART-B only. | |

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CSL31 | CO1 | Apply OOP Concepts to develop Programs |
| | CO2 | Demonstrate design of applications using Class and Objects |
| | CO3 | Adopt Inheritance feature to develop programs. |
| | CO4 | Illustrate Reusability using Function and Operator overloading |
| | CO5 | Create dynamic Objects using virtualization |

| **Course Title: Data Structures Lab** | | |
|---|---|---|
| Subject Code : **19CSL32** | Credits : 1 | CIE: 50 |
| Number of Lecture Hours/Week | **3 Hrs (Practical)** | SEE: 50 |
| | | SEE Hours: 03 |
| **Prerequisite:** C Language : Functions and Pointers | | |

**Course Objectives :**

1. To study the working of data structures such as stacks, queues, trees, hash tables, search trees.
2. To choose the appropriate data structure for a specified application.
3. To learn various searching and sorting algorithms.

1. Design, Develop and Implement a menu driven Program in C for the following Array operations
   a. Creating an Array of N Integer Elements
   b. Display of Array Elements with Suitable Headings
   c. Inserting an Element (ELEM) at a given valid Position (POS)
   d. Deleting an Element at a given valid Position(POS)
   e. Exit.

   Support the program with functions for each of the above operation.

2. Design, Develop and Implement a program in C for the following operations on Strings
   a. Read a Main String (STR), a Pattern String (PAT) and a Replace String (REP).
   b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Repost suitable messages in case PAT does not exist in STR.

   Support the program with functions for each of the above operations. Don‟t use built-in functions.

3. Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)
   a. Push an Element on to Stack
   b. Pop an Element from Stack
   c. Display the status of Stack
   d. Demonstrate Overflow and Underflow situations on Stack
   e. Exit

   Support the program with appropriate functions for each of the above operations.

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, %( Remainder), ^ (Power) and alphanumeric operands.

5. Design, Develop and Implement a Program in C for the following Stack Applications
   a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^
   b. Solving Tower of Hanoi problem with n disks

6. Design, Develop and Implement a menu driven Program in C for the following operations on QUEUE of Characters (Array Implementation of Queue with maximum size MAX)
   a. Insert an Element on to QUEUE
   b. Delete an Element from QUEUE
   c. Demonstrate Overflow and Underflow situations on QUEUE
   d. Display the status of QUEUE
   e. Exit

   Support the program with appropriate functions for each of the above operations.

7. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of integer values
   a. Create a SLL of N integers by using front insertion.
   b. Display the status of SLL and count the number of nodes in it
   c. Perform Insertion and Deletion at End of SLL
   d. Perform Insertion and Deletion at Front of SLL

8. Design, Develop and Implement Program in C to Reverse a Singly Linked List (SSL) of a given integers.

9. Design, Develop and Implement a menu driven Program in C for the following operations on Priority Queue.
   a. Create a Priority queue by using Insert function.
   b. Insertion data and Priority values as Input.
   c. Perform Deletion operation.
   d. Display the elements of Priority queue.

| | |
|---|---|
| 10. Design, Develop and Implement a Program in C for the following operations on Binary Search Tree(BST) of Integers<br>a. Create a BST of N integers: 6,9,5,2,8,15,24,14,7,8,5,2.<br>b. Traverse the BST in Inorder<br>c. Traverse the BST in Preorder<br>d. Traverse the BST in Postorder<br>11. Given a File of N employee records with a set K of Keys(4- digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table(HT) of m memory locations with L as the set of memory addresses (2- digit) of locations in HT. Let the keys in K and Addresses in L are Integers. Design and develop a Program in C that uses Hash function H: K ®L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing. | |

**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CSL32 | CO1 | Design and develop various data structure using pointers , dynamic memory allocation and recursion |
| | CO2 | Demonstrate basic operations on linked list using suitable data structures. |
| | CO3 | Illustrate the implementation of different sorting and searching techniques. |
| | CO4 | Construct Binary trees and binary search trees |
| | CO5 | Write a well organized laboratory report presenting the results in a clear way using algorithms and obtained output. |

| **Course Title: Analog and Digital Electronics Lab** | | |
|---|---|---|
| Subject Code : **19CSL33** | Credits : 1 | CIE: 50 |
| Number of Lecture Hours/Week | **3Hrs (Practical)** | SEE: 50 |
| | | SEE Hours: 03 |
| **Prerequisite:** Knowledge of Basic Electronics and Boolean algebra. | | |
| **Course Objectives :** | | |
| | | |

**Laboratory Experiments:**

1. Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.
2. Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.
3. Design and implement an Astablemultivibrator circuit using 555 timer for a given frequency and duty cycle.
4. Design and implement Half adder, Full Adder, Half Subtractor, Full Subtractor using basic gates.
5. Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
6. Design and implement code converter I)Binary to Gray (II) Gray to Binary Code using basic gates.
7. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.
8. Realize a D,T,JK Flip-Flop using NAND gates and verify its truth table.
9. Design and implement a mod-n (n<8) synchronous up counter using JK Flip Flop ICs and Demonstrate its working
10. Design and implement an Asynchronous counter using decade counter IC to count from 0 to n(n<=9) and demonstrate on seven segment display(using IC 7447)
11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).
12. To study 4-bitALU using IC-74181.

**Question paper pattern:**
**Note:** Conduction of Practical Examination: All laboratory experiments (1 to 11 nos) are to be included for practical examination.

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CSL33 | **CO1** | Demonstrate various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit. |
| | **CO2** | Evaluate and demonstrate various combinational logic circuits. |
| | **CO3** | Create and demonstrate various types of counters and Registers using Flip-flops. |
| | **CO4** | Develop D/A converters. |
| | **CO5** | Analyze the working and implementation of ALU. |

# CURRICULUM
# FOR THE ACADEMIC YEAR 2020-2021

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## B.E. IV SEMESTER


**POOJYA DODDAPPA APPA COLLEGE OF ENGINEERING**
**(An autonomous college under VTU)**
**KALABURAGI**

**About the institution:** The Hyderabad Karnataka Education (HKE) society founded by Late Shri Mahadevappa Rampure, a great visionary and educationist.The HKE Society runs 46 educational institutions. Poojya Doddappa Appa College of Engineering, Gulbarga is the first institution established by the society in 1958. The college is celebrating its golden jubilee year, setting new standards in the field of education and achieving greater heights. The college was started with 50% central assistance and 50% state assistance, and a desire to impart quality technical education to this part of Karnataka State. The initial intake was 120 with degree offered in three branches of engineering viz, Civil, Mechanical and Electrical Engineering. Now, it houses 11 undergraduate courses, 10 post Graduate courses and 12 Research centers, established in Civil Engg., Electronics & Communication Engg, Industrial & Production Engg, Mechanical Engg, Electrical Engg., Ceramic Cement Tech., Information Science & Engg.,Instrumentation Technology, Automobile Engg., Computer Sc. and Engg., Mathematics and Chemistry All the courses are affiliated to Visveswaraya Technological University, Belgaum. At present the total intake at UG level is 980 and PG level 193.

The college receives grant in aid funds from state government. A number of projects have been approved by MHRD /AICTE, Govt. of India for modernization of laboratories. KSCST, Govt. of Karnataka is providing financial assistance regularly for the student's projects.

The National Board of Accreditation, New Delhi, has accredited the College in the year 2005-08 for 09 UG Courses out of which 08 courses are accredited for three years and 01 course is accredited for five years. And second time accredited for Six Course in the year 2009-2012

Our college is one among the 14 colleges selected under TEQIP, sponsored by World Bank. It has received a grant of Rs 10.454 Crores under this scheme for its development. The institution is selected for TEQIP phase II in year 2011 for four years. Institution is receiving a grant of Rs 12.50 Crores under TEQIP Phase -II scheme for its development and selected for TEQIP-III as mentoring Institute for BIET Jhansi(UP).

Recognizing the excellent facilities, faculty, progressive outlook, high academic standards and record performance, the VTU Belgaum reposed abundant confidence in the capabilities of the College and the College was conferred Autonomous Status from the academic year 2007-08, to update its own programme and curriculum, to devise and conduct examinations, and to evaluate student's performance based on a system of continuous assessment. The academic programmers are designed and updated by a Board of Studies at the department level and Academic Council at the college level. These statutory bodies are constituted as per the guidelines of the VTU Belgaum. A separate examination section headed by a Controller of Examinations conducts the examinations.

At present the college has acquired the Academic autonomous status for both PG and UG courses from the academic year 2007-08 and it is one among the six colleges in the state of Karnataka to have autonomous status for both UG and PG courses.

One of the unique features of our college is, it is the first college in Karnataka State to start the Electronics and Communication Engineering branch way back in the year 1967, to join NIT Surathkal and IISc, Bangalore. Also, it is the only college in the state and one among the three

colleges across the country, offering a course in Ceramic and Cement Technology. This is the outcome of understanding by faculty and management about the basic need of this region, keeping in view of the available raw material and existing Cement Industries.

Bharatiya Vidya Bhavan National Award for an Engineering College having Best Overall Performance for the year 2017 by ISTE (Indian Society for Technical Education). In the year 2000, the college was awarded as Best College of the year by KSCST, Bangalore in the state level students projects exhibition.

The college campus is spread over 71 acres of land on either side of Mumbai-Chennai railway track and has a sprawling complex with gardens and greenery all around.

**About the department:** The Computer Science and Engineering department was started in the year 1984 with an intake of 40 students for UG. The department has seen phenomenal growth and now the department has increased UG intake to 120 students and offering two Post Graduation programmes : PG (Computer Science and Engineering with an intake of 25 students) and PG(Computer Network and Engineering with an intake of 18 students). The department is offering research program under its recognized research center. The department is having state-of-the-art computing facilities with high speed internet facilities and laboratories. The department library provides useful resources like books and journals. The department has well qualified and experienced teaching faculty. The department has been conducting several faculty development programs and student training programs.

**Vision of the institute:**

To be an institute of excellence in technical education and research to serve the needs of the industry and society at local and global levels.

**Mission of the institute:**

- Toprovideahigh-qualityeducationalexperienceforstudentswithvaluesandethicsthatenablesthemtobecomeleadersintheirchosenprofessions.
- To explore, create and develop innovations in engineering and science through research and development activities.
- Toprovidebeneficialservicetothenationalandmultinationalindustriesandcommunitiesthroughheducational, technical and professional activities.

**Department Vision**

To become pioneers in computer education and research and to prepare highly competent IT professionals to serve Industry and Society at local and global levels.

**Department Mission**

- To impart high quality professional education to become a leader in Computer Science and Engineering.
- To achieve excellence in Research for contributing to the development of the society.
- To inculcate professional and ethical behavior to serve the Industry.

**Program Educational Objectives (PEOs) of the department**

**PEO1**: To prepare graduates with core competencies in mathematical and engineering fundamentals to solve and analyze computer science and engineering problems.

**PEO2**: To adapt to evolving technologies and tools for serving the Society.

**PEO3**: To perform as team leader, effective communicator and socially responsible computer professional in multidisciplinary fields following ethical values.

**PEO4**: To encourage students to pursue higher studies and engage in research and entrepreneurship.

**Program Specific Outcomes (PSOs) of the department**

**PSO1**: Acquire competency in hardware and software working principles to analyze and solve computing problems.

**PSO2**: Design quality software to develop scientific and business applications following Software Engineering practices.

**PSO3**: Apply cutting edge technologies using modern tools to find novel solutions ethically to existing problems.

**Program Outcomes**

Engineering Graduates will be able to:

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

2**. 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.

3. **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.

4. **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.

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

6. **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.

7. **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.

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

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

10. **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.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as member and leader in a team, to manage projects and in multidisciplinary environments.

12. **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.

## SCHEME OF TEACHING FOR IV SEMESTER -2020-2021

| Code No. | Course | Hours/Week | | | | Maximum Marks | | |
|---|---|---|---|---|---|---|---|---|
| | | Lecture | Tutorial | Practical | Credits | CIE | SEE | Total |
| **SEMESTER IV** | | | | | | | | |
| 19CS41 | Applied Statistics | 2 | 2 | -- | 3 | 50 | 50 | 100 |
| 19CS42 | Operating System | 4 | -- | -- | 4 | 50 | 50 | 100 |
| 19CS43 | Java Programming | 4 | -- | -- | 4 | 50 | 50 | 100 |
| 19CS44 | Analysis & Design of Algorithm | 3 | 2 | -- | 4 | 50 | 50 | 100 |
| 19CS45 | Microprocessor and Microcontroller | 3 | 2 | -- | 4 | 50 | 50 | 100 |
| 19CS46 | Environment Studies | 2 | -- | -- | 0 | 50 | 50 | 100 |
| 19KAN47 | Kannada | 1 | -- | -- | 1 | 50 | 50 | 100 |
| 19CSL41 | Analysis & Design of Algorithm Lab | 0 | 0 | 3 | 1 | 50 | 50 | 100 |
| 19CSL42 | Microprocessor and Microcontroller Lab | 0 | 0 | 3 | 1 | 50 | 50 | 100 |
| 19CSL43 | Java Programming Lab | 0 | 0 | 3 | 1 | 50 | 50 | 100 |
| | Total | **19** | **6** | **9** | **23** | **500** | **500** | **1000** |

| **Course Title: Applied Statistics** | | |
|---|---|---|
| Subject Code : **19MA41** | Credit : 03 | CIE: 50 |
| Number of Lecture Hours/Week | **2 (Theory) + 2 (Tut)** | SEE: 50 |
| Total Number of Lecture Hours | 42 | SEE Hours: 03 |

**Prerequisites:** Basic knowledge of Statistic and Probability

**Course Objectives:** To enable the students to obtain the knowledge of Engineering Mathematics in the following topics

1.Probability distribution of discrete and continuous random variables
2.Joint probability distributions and discrete and continuous random variables and Morkov's chains
3.Analyse the sample data using Large sample test, t-distribution and chi- distribution

| **MODULES** | **Teaching Hours** |
|---|---|
| **Module I**<br><br>**Probability distributions:**<br>Random variable (Discrete and continuous) p.d.f., c.d.f., Binomial distribution, Poisson distributions, Normal distribution and problems | **8 hours** |
| **Module II**<br><br>**Joint probability distributions:**<br>Concept of joint probability distribution, discrete and continuous random variables independent random variables .problems on expectation and variance | **8 hours** |
| **Module III**<br><br>**Markov chains:**<br>**Introduction probability vectors** stochastic matrices, higher transition probability. Stationary distribution of regular Markov chains and absorbing states | **8 hours** |
| **Module IV**<br><br>**Sampling theory:**<br>Sampling, sampling distribution, standard error. Testing of hypothesis for means. Confidence limits for means. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations. Test of significance Small samples student's t-distribution: Test for single mean, difference of means, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes. And problems**.** | **9 hours** |
| **Module V**<br><br>**Distances in Classification:**<br>Introduction, Euclidean Distance, Manhattan Distance, Euclidean vs Manhattan Distance, Chebyshev Distance, Hamming Distance, Distance calculation in Clusters | **8 hours** |

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module**.**

**TEXT BOOKS:**

1 Higher Engineering Mathematics by B.S.Grewal, 36th Edn.

2 Engineering Mathematics by N. P. Bali and Manish Goyal. Laxmi publications, latest edition.

3 Higher Engineering Mathematics by H. K. Dass and Er. Rajnish Verma. S. Chand publishing 1st edition -2011

**REFERENCES:**

1. Advanced Engineering Mathematics by E. Kreyszig, John Willey & sons 8th Edn.
2. Advanced Engineering Mathematics by R.K.Jain & S.R.K Iyengar; Narosa publishing House.
3. Introductory methods of numerical analysis

**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19MA41 | CO1 | Solve problems using theoretical probability distributions |
| | CO2 | Apply the concepts of joint probability, to find covariance, correlation, independent variables |
| | CO3 | Apply stochastic to find the probability vectors, stochastic matrices and higher transition probability |
| | CO4 | Analyse the sample data using Large sample tests |
| | CO5 | Analyse the sample data using t-distribution and chi- distribution . |

| **Course Title: OPERATING SYSTEM** | | |
|---|---|---|
| Subject Code : **19CS42** | Credit : 4 | CIE: 50 |
| Number of Lecture Hours/Week | **4 hrs(Theory)** | SEE: 50 |
| Total Number of Lecture Hours | 52 | SEE Hours: 03 |
| **Prerequisites: Computer Organization, Microprocessor** | | |

**Course Objectives:**

- Learn services provided by the operating system and design of operating system
- Gain knowledge on how processes are synchronized and scheduled how different resources are managed.
- Understand structure and organization of file system and approaches to memory management.

| **MODULES** | **Teaching Hours** |
|---|---|
| **Module – I**<br>**Introduction:** What Operating Systems Do, Computer-System Organization, Computer-System Architecture, Operating-System Operations, Process Management, Memory Management, Storage Management, Security and Protection, Kernel Data Structures, Computing Environments.<br>**Operating-System Structures:** Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating System Structure.<br><br>**Case Studies:** Architecture of UNIX, The Kernel of Unix; The Kernel of Solaris; Architecture of Windows. | **10 Hrs** |
| **Module-II**<br>**Process Management**: Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication, Communication in Client–Server Systems.<br>**Multithreaded Programming:** Overview, Multicore Programming, Multithreading Models, Thread Libraries, Implicit Threading, Threading Issues | **10 Hrs** |
| **Module – III**<br>**Process Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multi-Processor Scheduling, Real-Time CPU Scheduling, Exercises<br>**Process Synchronization:** The Critical-Section Problem, Peterson's Solution, Synchronization hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors . | **11 Hrs** |
| **Module – IV**<br>**Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Exercises.<br><br>**Memory Management:** Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table, Example: Intel 32- and 64-bit Architectures, Example: ARM Architecture, Exercises. | **10 Hrs** |

| Module – V | 11 Hrs |
|---|---|
| **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory mapped files, Allocating Kernel Memory, Exercises.<br><br>**File System: File-System Interface:** File Concept, Access Methods, Directory and disk Structure, File system Mounting, File Sharing, and Protection | |

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module**.**

**Text book :**

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, 9th Edition, Wiley-India, 2018.
2. D.M Dhamdhere, Operating systems - A concept based Approach, 3rd Edition, Tata McGraw- Hill, 2012.

**Reference Books:**

1. P.C.P. Bhatt: Operating Systems, 2nd Edition, PHI, 2006.
2. Harvey M Deital: Operating systems, 3rd Edition, Addison Wesley, 2003.

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CS42 | **CO1** | Describe the functions of operating systems and its structures |
| | **CO2** | Describe process concepts and management models. |
| | **CO3** | Apply Scheduling algorithms and different concurrency control techniques to provide co-ordination among processes for the global data. |
| | **CO4** | Apply deadlock detection and prevention algorithms and memory management including the concept of paging, segmentation and swapping policies. |
| | **CO5** | Discuss Virtual memory management and describe file system interface. |

| **Course Title: Java Programming** | | |
|---|---|---|
| Subject Code : **19CS43** | Credit : 04 | CIE: 50 |
| Number of Lecture Hours/Week | 04 | SEE: 50 |
| Total Number of Lecture Hours | 52 | SEE Hours: 03 |
| **Prerequisites:** Concepts of Object oriented programming | | |
| **Course Objectives:** Learn the Java Programming to develop applications, creating GUI with applets, swings and web applications using servlets. | | |

| **MODULES** | **Teaching Hours** |
|---|---|
| **Module I**<br>**An Overview of Java** -Object-Oriented Programming, Two Paradigms Abstraction, The Three OOP Principles, A First Simple Program , Entering the Program, Compiling the Program ,A Closer Look at the First Sample Program , A Second Short Program, The Java Keywords , The Java Class Libraries,<br>**Data Types, Variables, and Arrays:** The Primitive Types - Integers, byte, short, int, long, Floating-Point Types, float, double, Characters , Booleans. A Closer Look at Literals - Integer Literals, Floating-Point Literals, Boolean Literals , Character Literals, String Literals, Variables, Declaring a Variable, Dynamic Initialization ,The Scope and Lifetime of Variables, Type Conversion and Casting, Java's Automatic Conversions, Casting Incompatible Types, Automatic Type Promotion in Expressions, The Type Promotion Rules , Arrays, One-Dimensional Arrays, Multidimensional Arrays, Alternative Array Declaration Syntax, A Few Words About Strings. Overview of Operators | **11 Hrs** |
| **Module II**<br>**Over view of Control Statements, Introducing Classes**-Class Fundamentals, The General Form of a Class, A Simple Class, Declaring Objects, A Closer Look at new, Assigning Object Reference Variables.<br>**Introducing Methods-** Adding a Method to the Box Class, Returning a Value, Adding a Method That Takes Parameters, Constructors, Parameterized Constructors, The this Keyword, Instance Variable Hiding, Garbage Collection, The finalize( ) Method, A Stack Class<br>**Inheritance-** Inheritance Basics, Member Access and Inheritance, A More Practical Example, A Superclass, Variable Can Reference a Subclass Object.<br>Using super: Using super to Call Superclass Constructors, A Second Use for super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding. | **10 Hrs** |
| **Module III**<br>**Dynamic Method Dispatch.** Why Overridden Methods? Applying Method Overriding, Using Abstract Classes, Using final with Inheritance, Using final to Prevent Overriding, Using final to Prevent Inheritance, The Object Class<br>**Packages and Interfaces:** Packages, Defining a Package, Finding Packages and CLASSPATH, A Short Package Example<br>**Access Protection:** An Access Example, Importing Packages,<br>Interfaces: Defining an Interface, Implementing Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces, Interfaces Can Be Extended. | |

| | |
|---|---|
| **Exception Handling:** Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Displaying a Description of an Exception, Multiple catch Clauses, Nested try Statements, throw, throws ,finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions. | **10 Hrs** |
| **Module IV**<br>**The Applet Class**-Two Types of Applets<br>**Applet Basics:** The Applet Class, Applet Architecture,<br>**An Applet Skeleton**: Applet Initialization and Termination, Overriding update( ), Simple Applet Display Methods,.<br>**Requesting Repainting:** A Simple Banner Applet, Using the Status Window , The HTML APPLET Tag.<br>**Passing Parameters to Applets:** Improving the Banner Applet, getDocumentBase( ) and getCodeBase( ), AppletContext and showDocument( ), The AudioClip Interface,The AppletStub Interface , Outputting to the Console.<br>**Event Handling**- Two Event Handling Mechanisms.<br>**The Delegation Event Model- Events:** Event Sources, Event Listeners.<br>**Event Classes:** The ActionEvent Class, The AdjustmentEvent Class, The ComponentEvent Class, The ContainerEvent Class,The FocusEvent Class,The InputEvent Class, The ItemEvent Class, The KeyEvent Class, The MouseEvent Class, The MouseWheelEvent Class, The TextEvent Class,The WindowEvent Class.<br>Using the Delegation Event Model- Handling Mouse Events, Handling Keyboard Events, Adapter Classes, Inner Classes, Anonymous Inner Classes.<br>**JDBC-ODBC Connectivity:** Talking to Database, Immediate Solutions, Essential JDBC program, using prepared Statement Object, Interactive SQL tool. JDBC in Action Result sets, Batch updates, Mapping, Basic JDBC data types, Advanced JDBC data types, immediate solutions. | **10 Hrs** |
| **Module V**<br>**Servlets -** Background, The Life Cycle of a Servlet, Using Tomcat for Servlet Development,.<br>**A Simple Servlet:** Create and Compile the Servlet Source Code , Start Tomcat , Start a Web Browser and Request the Servlet, The Servlet API,<br>**The javax servlet Package:** The Servlet Interface, The Servlet Config Interface, The Servlet Context Interface, The ServletRequest Interface, The Servlet Response Interface,The GenericServlet Class, The Servlet Input Stream Class, The ServletOutputStream Class, The Servlet Exception Classes, Reading Servlet Parameters.<br>**The javax,servlet.http Package:** The Http Servlet Request Interface,The Http Servlet Response Interface, The Http Session Interface, The Http Session Binding Listener Interface, The Cookie Class, Contents, The Http Servlet Class, The Http Session Event Class, The Http Session Binding Event Class,<br>**Handling HTTP Requests and Responses, Handling**: HTTP GET Requests,<br>Handling HTTP POST Requests, Using Cookies, Session Tracking. | **11 Hrs** |

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module**.**

**Text Books:**

1. Herbert Schildt , The Complete Reference, JAVA 7th/9th Edition, Tata McGraw Hill, 2013.
2. Java 6 Programming Black Book, Dreamtech Press. 2012

**Reference Books:**

1. Java Fundamentals: A comprehensive Introduction by Herbert Schildt, Dale Skrien. Tata McGraw Hill Edition 2013.
2. Stephanie Bodoff et al: The J2EE Tutorial, 2nd Edition, Pearson Education,2004.
3. Uttam K Roy, Advanced JAVA programming, Oxford University press, 2015.

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CS43 | CO1 | Apply the concepts of programming and implement programs using Java constructs. |
| | CO2 | Create classes and demonstrate object oriented programming concepts |
| | CO3 | Demonstrate inheritance , overloading and run-time errors using exception handling mechanism. |
| | CO4 | Develop GUI application program using Applet , event handling and database connectivity. |
| | CO5 | Design and develop web application using servelet programming. |

| **Course Title: ANALYSIS AND DESIGN OF ALGORITHMS** | | |
|---|---|---|
| Subject Code : **19CS44** | Credit : 4 | CIE: 50 |
| Number of Lecture Hours/Week | **03 Hrs+ 2Hrs(Tut)** | SEE: 50 |
| Total Number of Lecture Hours | **52** | SEE Hours: 03 |

**Pre-requisites:** Data structures using C.

**Course objectives:**

- Analyze the asymptotic performance of the algorithms in time and space domain.
- Introduce various algorithm design techniques.

| **Modules** | **Teaching Hours** |
|---|---|
| **Module-I**<br>Algorithm, Fundamentals of Algorithmic Problem Solving, Important problem Types, Fundamental of Data Structures, Fundamentals of the Analysis of Algorithm Efficiency; Analysis Framework, Asymptotic Notations , Basic Efficiency Classes, Non-recursive and Recursive Algorithms, Examples-Fibonacci Numbers. | **10 Hrs** |
| **Module-II**<br>**Brute Force:** Introduction, Selection sort, Bubble Sort, Sequential search Brute-Force String Matching Exhaustive Search<br>**Divide & Conquer :** Introduction, Merge Sort, Quick Sort, Binary Search, Binary tree traversals & related properties, Multiplication of large integers & Stressen's Matrix Multiplication Insertion Sort. | **11 Hrs** |
| **Module-III**<br>**Decrease & Conquer** : Introduction, Depth First search, Breadth First Search, Topological Sorting, Algorithms for Generating Combinatorial objects.<br>**Transform & Conquer :** Introduction , Presorting, Balanced Search Trees, 2-3 Trees, Heaps and Heap Sort, Problem Reduction, **Space & Time Tradeoffs :** Sorting by Counting, Input Enhancement in String matching , Hashing. | **11 Hrs** |
| **Module-IV**<br>**Dynamic Programming:** Introduction, Computing a Binomial Coefficient, Warshall's Algorithm, Floyd's Algorithm, The Knapsack Problem and Memory Functions.<br>**Greedy Techniques**: Introduction, Minimum Spanning Tree, Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffmancodes. | **10 Hrs** |
| **Module-V**<br>**Limitations of Algorithms Power:** Introduction, Lower- Bound Arguments, Decision Trees, P, NP, and NP – Complete Problems.<br>**Backtracking:** Introduction, n-Queen‟s problem, Hamiltonian circuit problem, Subset problem, General backtracking algorithm, **Branch- and-Bound :** The assignment problem , Knapsack problem, Travelling sales man problem. | **10 Hrs** |

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text books:**

1. Anany Levitin, " Introduction to the Design & Analysis of Algorithm ", 2nd Edition, Pearson Edition, 2007.

**Reference Books:**

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, "Introduction Algorithm", 2nd Edition, PHI,2006.
2. Horowitz E, Sahni S., Rajasekaran S., "Computer Algorithms", Galgotia Publications,2001.

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CS44 | **CO1** | Explain fundamental ideas used for designing and analyzing algorithms. |
| | **CO2** | Demonstrate Brute Force, Divide-and-Conquer techniques and analyze the performance of algorithms. |
| | **CO3** | Demonstrate design of Decrease & Conquer and Transform & Conquer algorithms and their efficiencies. |
| | **CO4** | Apply Dynamic Programming and Greedy Techniques to solve various graph problems efficiently. |
| | **CO5** | Describe Limitations of algorithms power and illustrate Back tracking, Branch-and-Bound algorithms to solve recursive and computational problems. |

| Course Title: **MICROPROCESSOR AND MICROCONTROLLER** | | |
|---|---|---|
| Subject Code : **19CS45** | Credit : 4 | CIE: 50 |
| Number of Lecture Hours/Week | **03 Hrs(Th) + 2Hrs(Tut)** | SEE: 50 |
| Total Number of Lecture Hours | 52 | SEE Hours: 03 |
| **Pre-requisites:** Logic Design, Basic Electronics | | |
| **Course objectives:**<br>• Explore the microprocessor architecture and its instruction set.<br>• Develop skills for programming in Assembly language.<br>• Interface Peripheral devices with 8086 Microprocessor and ARM Processor | | |
| **Modules** | | **Teaching Hours** |
| **Module-I**<br>**Basic Structures of Computers, machine instructions & programs:** Basic Operational Concepts : Bus Structures, Performance, Processor clock, Basic Performance equation, Pipelining & Superscalar operation, Clock rate, Performance measurement, Multiprocessor & Multicomputer.<br>The x86 microprocessor: Inside the 8088/86, Introduction to assembly programming, Introduction to Program Segments, The Stack, Flag register, x86 Addressing Modes. Assembly language programming: Directives & a Sample Program, Assemble, Link & Run a program, More Sample programs, Control Transfer Instructions, Data Types and Data Definition, Full Segment Definition. | | **12 Hrs** |
| **Module-II**<br>**x86:** Instructions sets description, Arithmetic and logic instructions and programs: Unsigned Addition and Subtraction, Unsigned Multiplication and Division, Logic Instructions, BCD and ASCII conversion, Rotate Instructions. INT 21H and INT 10H Programming : Bios INT 10H Programming , DOS Interrupt 21H. 8088/86 Interrupts, x86 PC and Interrupt Assignment. | | **11 Hrs** |
| **Module-III**<br>**Signed Numbers and Strings:** Signed number Arithmetic Operations, String operations. Memory and Memory interfacing: Memory address decoding, data integrity in RAM and ROM, 16-bit memory interfacing. 8255 I/O programming: I/O addresses MAP of x86 PC‟s, programming and interfacing the 8255. | | **10 Hrs** |
| **Module-IV**<br>**Microprocessors versus Microcontrollers, ARM Embedded Systems:** The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software, ARM Processor Fundamentals : Registers , Current Program Status Register , Pipeline, Exceptions, Interrupts, and the Vector Table , Core Extensions | | **10 hrs** |

| Module-V | |
|---|---|
| **Introduction to the ARM Instruction Set :** Data Processing Instructions , Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants, Simple programming exercises. | **09 Hrs** |
| | |

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module, covering all the topics from a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text books:**

1. Carl Hamacher, Z.Vranesic & S.Zaky, Computer Organization, 5th Edition , Tata McGraw-Hill Publishing Company Ltd. New Delhi, 2011.
2. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Danny Causey, The x86 PC Assembly Language Design and Interfacing, 5th Edition, Pearson,2013.
3. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers,2008.

**Reference Books:**

1. Douglas V. Hall: Microprocessors and Interfacing, Revised 2nd Edition, TMH,2006.
2. K. Udaya Kumar & B.S. Umashankar : Advanced Microprocessors & IBM-PC Assembly Language Programming, TMH 2003.
3. Ayala : The 8086 Microprocessor: programming and interfacing - 1st edition, CengageLearning
4. Joseph Yiu, The Definitive Guide to the ARM Cortex-M3, 2nd Edition , Newnes,2009
5. The Insider‟s Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition,2005
6. Steve Furber, ARM System-on-Chip Architecture, Second Edition, Pearson,2015
7. Architecture, Programming and Interfacing of Low power Processors- ARM7, Cortex-M and MSP430, Lyla B Das Cengage Learning, 1stEdition

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CS45 | CO1 | Analyze the 8086 processor Structure, Assembly Language Programming and System programs used in Assembly language programming. Acquire knowledge on basic structure of computer and its performance |
| | CO2 | Develop assembly language code to solve problems |
| | CO3 | Design hardware interfacing of memory devices to x86 family |
| | CO4 | Compare Microprocessor and Microcontroller, Explain interfacing through ARM processor, interrupt routines |
| | CO5 | Demonstrate Instruction set and develop programs using ARM processor |

| Course Title: ANALYSIS AND DESIGN OF ALGORITHMSLAB | | |
|---|---|---|
| Subject Code : **19CSL41** | Credits : 1 | CIE: 50 |
| Number of Lecture Hours/Week | **3 Hrs (Practical)** | SEE: 50 |
| | | SEE Hours: 03 |

**Prerequisite:** C Language : Functions and Recursion

**Corse Objectives :** To **e**nable the students for

- Learn different searching and sorting techniques.
- Gain knowledge of binary tree principles.
- Understand the different algorithms to solve the problems.

**PART – A**

**Using C / C++**

1. Write a C Program to implement Recursive Binary search and linear search and determine the time required to search an element.

2. Write a C Program to sort a given set of elements using Merge sort method and determine the time required to sort the elements.

3. Write a C Program to Sort a given set of elements using Selection sort and determine the time required to sort elements.

4. Write a C Program to Sort a given set of elements using Insertion sort and determine the time required to sort elements.

5. Write a C Program to Sort a given set of elements using the Heap sort method and determine the time required to sort the elements.

6. Write a C Program to Sort a given set of elements using Quick sort method and determine the time required sort the elements.

7. Write a C Program to Print all the nodes reachable from a given starting node in a digraph using BFS method.

8. Write a C Program to Check whether a given graph is connected or not using DFS method.

**PART – B**

1. Write a C Program to Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

2. Write a C Program to Find Minimum Cost Spanning Tree of a given undirected graph using Prim‟s algorithm.

3. Write a C Program to From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra'salgorithm.

4. Write a C Program to implement 0/1 Knapsack problem using dynamic programming.

5. Write a C Program to Find a subset of a given set S = {sl,s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2,5,6, 8} and d = 9 there are two solutions{1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.

6. Write a C Program to Implement Horspool algorithm for String Matching.

7. Write a C Program to Find the Binomial Co-efficient using Dynamic Programming.
8. Write a C Program to Implement Floyd‟s algorithm for the All-Pairs-Shortest- Paths problem.
9. Write a C Program to Compute the transitive closure of a given directed graph using Warshall‟s algorithm.

10. Write a C Program to Implement N Queen's problem using Back Tracking.

**Question paper pattern:**

**Note : For SEE, students will be asked to execute two programs, selecting one program from each part.**

**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CSL41 | CO1 | Apply the knowledge of Divide-and-Conquer techniques for different searching and sorting problems using recursive method and find the time complexity of algorithms. |
| | CO2 | Demonstrate Decrease-and-Conquer techniques for solving the graph problems. |
| | CO3 | Design and implement algorithms for solving the graph problems by using Greedy techniques. |
| | CO4 | Demonstrate the concepts of Dynamic Programming techniques by calculating the Binomial Co-efficient. |
| | CO5 | Illustrate the Back Tracking algorithms for subset and N-Queen‟s problems. |

| **Course Title: MICROPROCESSOR AND MICROCONTROLLER LAB** | | |
|---|---|---|
| Subject Code : **19CSL42** | Credits : 1 | CIE: 50 |
| Number of Lecture Hours/Week | **3 Hrs (Practical)** | SEE: 50 |
| | | SEE Hours: 03 |
| **Prerequisite:** C Programming | | |
| **Course Objectives : Explore the microprocessor architecture and instruction set** | | |

| **List of Programs** | |
|---|---|
| SOFTWARE PROGRAMS: PART A<br><br>1. Design and develop an assembly language program to search a key element "X" in a list of „n" 16-bit numbers. Adopt Linear search algorithm in your program for searching.<br><br>2. Design and develop an assembly program to sort a given set of „n" 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.<br><br>3. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriatemessage.<br><br>4. Develop an assembly language program to compute nCr using recursive procedure. Assume that „n" and „r" are non-negativeintegers.<br><br>5. Design and develop an assembly language program to read the current time and Date from the system and display it in the standard format on thescreen.<br><br>HARDWARE PROGRAMS: PART B<br><br>6. a. Design and develop an assembly program to demonstrate BCD Up-Down Counter (00-99) on the Logic Controller Interface.<br>b. Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.<br><br>7. Design and develop an assembly program to display messages "FIRE" and "HELP" alternately with flickering effects on a 7-segment display interface for asuitableperiodoftime.Ensureaflashingratethatmakesiteasytoreadboth | |

| | |
|---|---|
| the messages (Examiner does not specify these delay values nor is it necessary for the student to compute these values).<br><br>8. Design and develop an assembly program to drive a Stepper Motor interface and rotate the motor in specified direction (clockwise or counter- clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student) using ARM TTDMI/LPC2148.<br><br>9. Design and develop an assembly language program to a. Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO). b. Generate a Half Rectified Sine waveform using the DAC interface. ) using ARM TTDMI/LPC2148.<br><br>10. To interface LCD with ARM processor-- ARM7TDMI/LPC2148. Write and execute programs in C language for displaying text messages and numbers on LCD.<br>Study Experiments:<br>1. Interfacing of temperature sensor with ARM freedom board (or any other ARM microprocessor board) and display temperature on LCD<br><br>2. To design ARM cortex based automatic number plate recognition system<br>3. To design ARM based power saving system | |

**Question paper pattern:**

Conduction of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.
- Software Required: Open source ARM Development platform, KEIL IDE and Proteus for simulation

**Course outcomes:**

**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CSL42 | CO1 | Develop ALP for fixed and Floating Point and Arithmetic operations using 8086 microprocessor. |
| | CO2 | Design and develop assembly programs using 8086 DOS functions in assembly language |
| | CO3 | illustrate how the different peripherals (8255, 8253 etc.) are interfaced with Microprocessor. |
| | CO4 | Design circuits for various applications using ARM microcontrollers |
| | CO5 | Construct different waveforms using interfacing 8086 microprocessor |

| **Course Title: Java Programming Laboratory** | | |
|---|---|---|
| Subject Code : **19CSL43** | Credit : 1 | CIE: 50 |
| Number of Lecture Hours/Week | 3 | SEE: 50 |
| Total Number of Lecture Hours | | SEE Hours: 03 |
| **Prerequisites: Concepts of Object Oriented Programming, Programming** | | |
| **Course Objectives:** Learn to code and execute Java programs to solve problems, design of GUI for Java applications, Servlets for web applications, and database connectivity. | | |

| **MODULES** | **Teaching Hours** |
|---|---|
| **Preliminary practice programs**:<br><br>i) Understand and acquaint with Eclipse IDE environment. Write and execute a Java program to store and access student information.<br><br>ii) Write and execute a Java program to calculate sum of series of natural numbers<br><br>iii) Write and execute a Java program to demonstrate the scope of variables.<br><br>iv) Write and execute a Java program to find the biggest name in the array of strings.<br><br>v) Write and execute a Java program to demonstrate data type casting.<br><br>**Regular Laboratory exercises (for SEE):**<br><br>**(Every program should be a separate project and a package in Eclipse IDE)**<br><br>1. Write and execute a JAVA program to demonstrate use of any five string functions. Use both parameterized and non-parameterized constructors for passing string inputs.<br><br>2. Write and execute a JAVA Program to demonstrate Inheritance.<br><br>3. Write and execute a JAVA Program to demonstrate exception handling (both built- in and user-defined exceptions).<br><br>4. Write and execute a JAVA Program to implement inheritance (single level and multilevel).<br><br>5. Write and execute a JAVA program to demonstrate polymorphism through method overloading.<br><br>6. Write and execute a JAVA program to demonstrate method overriding. | |

| | |
|---|---|
| 7. Write a JAVA applet program and required HTML file to create banner applet.<br><br>8. Write a JAVA applet program to create a basic Applet having buttons, text area GUI controls to add & subtract two nos. Use appropriate event listeners.<br><br>9. Write a Java program to store, delete and update data in a database with the support of JDBC-ODBC connectivity.<br><br>10. Write a Java program with Servlets to store only valid data in a database with the support of JDBC-ODBC connectivity.<br><br>11. Write a JAVA Servlet program to create login page for authentication purpose.<br><br>12. Write a JAVA Servlet program for adding cookies to HTML page for counting number of visits to a web page. | |

**Question paper pattern:**
**Note : For SEE, students will be asked to execute two programs, selecting one program from each part.**

**REFERENCES:**
www.tutorialpoint.com , www.w3schools.com

**Course outcomes:**
**On completion of the course, the student will have the ability to:**

| Course Code | CO # | Course Outcome (CO) |
|---|---|---|
| 19CSL43 | CO1 | Implement Java programs with basic concepts of Object oriented programming. |
| | CO2 | Demonstrate Run-time and user-defined exceptions, Constructors. |
| | CO3 | Develop code for Inheritance, method overriding and overloading |
| | CO4 | Design interactive GUI Java programs using applets and database connectivity |
| | CO5 | Create Servlet for web applications |