

SYLLABUS OF

Bachelor of Science in Information Technology

PROGRAM



**Department of Computer Science
Gauhati University**

Program Structure

Semester	Paper Name	Course type	Credit
I	Computer Fundamentals	Compulsory	4 (3+1)
	Introduction to C-Programming	Compulsory	4 (3+1)
	Mathematical Foundation in Information Technology	Compulsory	4
II	Data Structures & Algorithms Using C	Compulsory	4(3+1)
	Digital Logic Fundamentals	Compulsory	4
	Computer Oriented Numerical and Statistical Methods	Compulsory	4 (3+1)
III	Computer Organization and Architecture	Compulsory	4
	Operating System	Compulsory	4(3+1)
	Object Oriented Programming through C++	Compulsory	4(3+1)
IV	Database Management System	Compulsory	4(3+1)
	Python Programming	Compulsory	4(3+1)
	Automata Theory and Languages	Compulsory	4
	Design and Analysis of Algorithms	Compulsory	4(3+1)
V	Software Engineering	Compulsory	4
	Web Technologies	Compulsory	4(3+1)
	Java Programming	Compulsory	4(3+1)
	Computer Networks	Compulsory	4(3+1)
VI	i) Computer Graphics	Elective I	4(3+1)
	ii) Information Security and Cyber Laws		4
	iii) System Software		4 (3+1)
	i) Artificial Intelligence	Elective II	4(3+1)
	ii) Advanced Web Programming		
	iii) Data Mining and Warehousing		
	i) Optimization Techniques	Elective III	4
	ii) Mobile Application Development		4(3+1)
	iii) Graph Theory		4(3+1)
	Project		4

Paper Name: Mathematical Foundation in Information Technology

1. Learning Outcomes: After successful completion of this course, students will be able to:

- a) Understand the Mathematical model of a finite state machine. Know deterministic and non-deterministic versions of Finite automata.
- b) Grasp the mathematical concepts of languages and grammar.
- c) Know Pushdown Automata and the associated grammar/language.
- d) Know the properties of Regular languages and Context free languages.

2. Prerequisites: NIL

3. Semester: 1

4. Course Type: Compulsory

5. Course Level: 100-199

6. Theory credit: 4

7. Practical credit: 0

8. Number of required hours:

- a) Theory: 60 hrs
- b) Practical: 0 hrs
- c) Non Contact: 5 hrs

9. List of books:

- a) Introduction to Automata Theory, Languages and Computation, Hopcroft, Motwani and Ullman, Pearson.
- b) Theory of Computer Science (Automata, Languages and Computation), K. L. P. Mishra, N. Chandrasekaran; P. H. I.

10. Contents of Syllabus:

UNIT 1:

(16

Lectures)

Sets, Relations and Functions

Sets: definition of set, cardinality of sets, finite, countable and infinite sets. Operations on sets, Venn diagram. Principle of inclusion and exclusion and their applications on simple problems. Multisets.

Relations: Definition and properties of binary relations, closures of relations, equivalence relations, equivalence classes and partitions, n-ary relations and representation of n-ary relations as tables. Partial ordering relations and lattices,

Functions: Definition of function, one-to-one and onto, principles of mathematical induction. Concave and convex functions.

UNIT 2: Combinatorics

(15lectures)

Basic of counting principles, principle of inclusion-exclusion, application of inclusion and exclusion, Mathematical Induction. Pigeonhole principle, generalized Pigeonhole principle and its application, permutations and combinations, circular permutations, permutations with repetitions, combinations with repetitions, permutations of sets with indistinguishable objects

UNIT 3: Growth of Functions

(5 Lectures)

Asymptotic behavior of functions, Asymptotic Notations - Big-O and Theta. Summation formulas and properties, Bounding Summations.

UNIT 4: Graph Theory

(12 Lectures)

Basic Definition of graph, Directed, Undirected and Weighted Graphs. Representation of graphs in Computers – Adjacency Matrix and Adjacency Lists. Degree of vertices – indegree and outdegree. Paths, Cycles and Acyclic graphs. Simple operations on graphs and amount of computations required for each operation. Connected graph, Tree and Forest. Bipartite graph, Algorithms on graph traversals- Breadth first search, Depth first search.

UNIT 5: Mathematical Logic

(12 Lectures)

Connectives, truth tables, Tautologies and Contradictions, Equivalence and Implications, NAND and NOR, Normal forms- CNF, DNF, Converting expressions to CNF and DNF, Theory of inference, Propositional Calculus, Predicate calculus (only introduction), predicates and quantifiers.

Particulars of Course Designer:

Name: Prof. Anjana Kakoti Mahanta

Contact No.: 9864425716

E-mail id : anjana@gauhati.ac.in

Paper Name: Computer Oriented Numerical and Statistical Methods

1. **Learning Outcome:** On successful completion of this course, the student should be able to:
 - learn the properties of Floating Point, Numbers and their accuracy, approximations and errors
 - Learn various probability methods, Interpolation methods etc.
 - To solve basic problems in probability and statistics
2. **Prerequisite:** NIL
3. **Semester:** 2
4. **Course Type:** Compulsory
5. **Course Level:** 100-199
6. **Theory credit:** 3
7. **Practical credit:** 1
8. **Number of required hours:**
 - d) Theory: 45 hrs
 - e) Practical: 30 hrs
 - f) Non Contact: 5 hrs
9. **List of books:**
 - a) Rajaraman, V, "Computer Oriented Numerical Methods", 3rd edition, Prentice Hall
 - b) Balaguruswami, E., "Computer Oriented Statistical and Numerical Methods", Macmillan Publishers India Limited

10. Detailed Syllabus:

a) Theory

UNIT-I: Introduction to Computer Arithmetic (7 hrs)

Representation of numbers: Fixed Point and Floating point representations, Normalized Floating Representation, Floating Point Arithmetic, Properties of Floating Point, Numbers and their accuracy, Approximations and errors. Errors: truncation error, rounded off error, absolute error, relative error, percentage error and error propagation

UNIT-II: Algebraic and Transcendental Equations (8 hrs)

Introduction to linear and nonlinear equations, measures of accuracy, Properties of polynomial equations, Initial approximation to a root, Solution of algebraic/transcendental equations: Bisection Method, Iteration method, Method of false position, Newton-Raphson method, Rate of convergence of Iterative methods, Solution of simultaneous linear equations by using Gauss elimination method

UNIT-III: Interpolation (6 hrs)

Polynomial Interpolation, Finite Differences, Newton's Forward Difference Interpolation, Newton's Backward Difference Interpolation, Newton's Divided Difference Interpolation

UNIT-IV: Solution of Differential Equation (6 hrs)

Taylor series method, Euler's method, Runge-Kutta method of 1st, 2nd & 4th order.

UNIT-V: Descriptive Statistics (6 hrs)

Types of Data, Attributes and Variables, Construction of Frequency, Cumulative frequency, Graphical Representation of Frequency distribution: Histogram, Frequency Polygon, Frequency Curve and

Cumulative Frequency Curves (Ogive curves), Diagrammatic Representations: Simple bar, Subdivided bar, Pie Diagrams

UNIT-VI: Measure of central tendency

(4 hrs)

Measure of central tendency-Mean, Median and Mode. Measure of variation-Range, Interquartile range, Standard Deviation and Variance

UNIT-VII: Measure of central tendency

(8 hrs)

Sample Space, events, random variables, Discrete probability, Conditional Probability and Bayes theorem, Linear Regression and Correlation, Probability Distribution Functions-Binomial, Random and Poisson

b) Lab Content:

Practical / Lab work to be performed using C/C++/Java programming Language:

- Apply the Bi-section method for approximation of root for a given polynomial equation.
- Apply the False Position method for approximation of root for a given polynomial equation
- Implement Newton Raphson method for approximation of root for a given polynomial equation.
- Implement Gauss elimination method to solve simultaneous linear equations
- Develop programs to implement Newton's Forward Difference Interpolation
- Develop programs to implement Newton's Backward Difference Interpolation
- Develop programs to implement Newton's Divided Difference Interpolation
- Develop program to apply Taylor's series for e raise to the power x
- Implement Euler's method for solving a differential equation
- Implement Runge-Kutta method of 1st, 2nd & 4th order for solving a differential equation
- Write programs to find Mean, Median and Mode for a given set of data

Particulars of course designer:

Name: Dwipen Laskar

Contact No.: 6000795681

E-mail id: laskardwipen@gauhati.ac.in

Paper Name: Data Structures & Algorithms Using C

1. **Learning Outcome:** At the end of the course, students will be able to:

- Understand and apply the fundamental data structures and algorithms – such as arrays, linked lists, stacks, queues, trees, sorting and searching algorithms using C programming language.
- Analyze the time and space complexity of different algorithms and choose the appropriate algorithm for a given problem.
- Develop efficient algorithms to solve various computational problems by utilizing data structures and algorithms covered in the course.

2. **Prerequisite:** NIL

3. **Semester:** 2

4. **Course Type:** Compulsory

5. **Course Level:** 100-199

6. **Theory credit:** 3

7. **Practical credit:** 1

8. **Number of required hours:**

- g) Theory: 45 hrs
- h) Practical: 30 hrs
- i) Non Contact: 5 hrs

9. **List of books:**

- a) Weiss, Mark Allen. “Data Structures and Algorithm Analysis in C”. 3rd ed., Pearson, 2012
- b) Sedgewick, Robert. “Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms”. 3rd ed., Addison-Wesley Professional, 2002.
- c) Goodrich, Michael T., and Roberto Tamassia. “Data Structures and Algorithms in C”. 2nd ed., Wiley, 2011.
- d) Gilberg, Richard F., and Behrouz A. Forouzan. “Data Structures: A Pseudocode Approach with C”. Narosa Publishing House, 2009.

10. **Detailed Syllabus:**

Unit 1: Data Structures Overview and Arrays

(8 Lectures)

Concepts of Data Types, Abstract Data Type, Data Structure, Fundamental and Derived Data Types. Importance of data structures. Array as a data structure (characteristics, advantages, disadvantages). Representation of arrays – single and multidimensional. Address calculation of array element using column and row major ordering. Address translation functions for one & two dimensional arrays. Insertion and deletion in arrays. Use of arrays for large number representation.

Unit 2: Linked Lists

(9 Lectures)

Initialization and implementation of structures. Structure and pointers. Self referential structure. Introduction to linked lists. Singly linked list, doubly linked list, circular linked list. Operations on lists – creation, insertion, deletion, traversal, merging and splitting. Array of structures and Structure of Arrays. Array of lists and List of lists.

Unit 3: Stacks and Queues

(9 Lectures)

Definition of Stack and Queue. Representation of stacks and queues using arrays and linked lists. Stack operations – push, pop. Queue operation – enqueue, dequeue. Circular Queue, Priority Queue, Conversion of infix arithmetic expression containing arithmetic operators and parenthesis to postfix and prefix expression. Evaluation of postfix expression.

Unit 4: Binary Trees

(8 Lectures)

Definition of Trees – General tree and Binary tree. Basic terminologies – parent, child, height, depth, leaf, node, internal nodes, external nodes. Brief concept of Forest, ordered trees, strictly binary tree, complete binary tree. Representation of trees using arrays and linked lists. Binary tree traversal methods – pre-order, in-order, post-order. Recursive and non-recursive algorithms for traversal methods. Binary search trees. Operation on BST – creation, insertion and deletion of a node. Definition and characteristics of threaded binary trees, multi-way search trees. Breadth First Search, Depth First Search. Min heap and Max heap.

Unit 5: Searching and Sorting

(6

Lectures)

Linear and binary search. Indexed search. Hashing. Hash Functions – division method, mid square method, folding. Conflict resolution – linear and quadratic probe. Sorting algorithms – Insertion sort, Selection sort, Bubble sort, Merge sort, Quick sort, Counting sort, Heap sort. In-place sorting and stable sorting.

Unit 6: Analysis of Algorithm and Complexity

(5 Lectures)

Complexity measures of an algorithm – Time and space complexity. Average case and worst case analysis. Asymptotic notation as a measure of algorithm complexity, O and θ notations. Analysis of sorting algorithms and Searching algorithms in terms of time and space complexity in best, average and worst case.

List of Practical

(This is a suggestive list only. Questions need not be restricted to this list. The practical are advised to be performed in Linux environment using C programming language.)

1. Write a program to declare an array and initialize the values according to the user. Now ask the user for a number n and return the n^{th} element from the array.
2. Write a program to implement array initialized with the numbers divisible by three up to 30. Write a function which accepts the array and return the positions of the even numbers in the array.
3. Implement linked list in a program by writing functions for the following:
 - a. Create a singly linked list of n nodes
 - b. Count the number of nodes in the list
 - c. Print the values of all the nodes
 - d. Add a node at first, last and k^{th} position in the linked list
 - e. Delete a node from first, last and k^{th} position
 - f. Search for an element in the list. If found, return the position of the node. If not found, return a negative value.

4. Write a program to implement doubly linked list.
5. Write a function to concatenate two linked lists.
6. Write a program to take a number k and split the linked list after k^{th} position.
7. Write a program to merge two sorted linked lists.
8. Write a program to implement list of lists.
9. Write a program to implement stack using array. Use push and pop operations on the array representation of the stack. Check whether the stack is full or empty.
10. Write a program to implement stack using linked list. Use push and pop operations on the stack by inserting nodes and deleting nodes from the linked list. Also check if the stack is full or empty.
11. Write a program to evaluate a simple postfix expression using stack.
12. Write a program to convert a decimal number into binary number using stack.
13. Write a program to implement queue using array. Add new elements to the queue and remove elements from the queue represented by array. Check whether the queue is full or empty.
14. Write a program to implement queue using linked list. Add new elements to the queue and remove elements from the queue represented by linked list. Also check whether the queue is full or empty.
15. Implement binary search and linear search algorithms on arrays.
16. Implement binary search tree using array by writing a program to:
 - a. Create a binary search tree using array
 - b. Print the prefix notation of the BST
 - c. Print the infix notation of the BST
 - d. Print the postfix notation of the BST
 - e. Search for an element in the BST
17. Implement binary search tree using linked list by writing a program to:
 - a. Create a binary search tree using linked list
 - b. Print the prefix notation of the BST
 - c. Print the infix notation of the BST
 - d. Print the postfix notation of the BST
 - e. Search for an element in the BST
18. Implement following sorting algorithms:
 - a. Bubble sort
 - b. Insertion sort
 - c. Selection sort
 - d. Counting sort

Particulars of Course Designer:

Name : Risheraj Baruah

Contact No. : +91 8486942427

Email id : rishirajbaruah@gauhati.ac.in

Paper Name: Digital Logic Fundamentals

1. Learning Outcome:

- After completing this course, students will have grasp of fundamental concepts of digital logic that will make their base to understand the concepts of computer architecture and organization

2. Prerequisite: NIL

3. Semester: 2

4. Course Type: Compulsory

5. Course Level: 100-199

6. Theory credit: 4

7. Practical credit: 0

8. Number of required hours:

- j) Theory: 60 hrs
- k) Practical: 0 hrs
- l) Non Contact: 5 hrs

9. List of books:

- e) Digital Logic and Computer Design, M. Morris Mano, Pearson India
- f) Digital Logic and Computer Organization, V. Rajaraman, T. Radhakrishnan, PHI Learning

10. Detailed Syllabus:

Unit I: Introduction to Binary Number System

10 hrs

Binary numbers, number base conversions, octal and hexa decimal numbers, 1's complement and 2's complement, representation of signed binary number: 1's complement, 2's complement and signed magnitude, subtraction with complements, arithmetic addition and subtraction of signed binary numbers, binary codes: BCD, Excess-3, error detection code: parity bit, error correction code: Hamming code, gray code, ASCII, EBCDIC, binary logic, logic gates: AND, OR, inverter, buffer, NAND, NOR, XOR and equivalence

Unit II: Boolean Algebra, Logic Gates and Integrated Circuits

15 hrs

Definition of boolean algebra, two valued boolean algebra, duality principle, theorems and postulates of boolean algebra, precedence of boolean operators, boolean expression and Venn diagram, boolean functions and truth tables, complement of a boolean function, minterms and maxterms, canonical forms of a boolean function, sum of minterms and its short notation, product of maxterms and its short notation, conversion between canonical forms, standard form of a boolean function, digital logic gates, integrated circuits and levels of integration, digital logic families

Unit III: Simplification of Boolean Functions

10 hrs

Map minimization method, two variable map, three variable maps, four variable map, five variable map, NAND and NOR implementation of boolean functions, don't-care conditions, tabulation method

Unit IV: Combinational Circuits**12 hrs**

Definition of combinational circuit, design procedure, half adder, full adder, half subtractor, full subtractor, BCD-to-Excess-3 code converter, encoders and decoders, multiplexers, ROM

Unit V: Sequential circuits**13 hrs**

Flip flops, RS flip flop, D flip flop, JK flip flop, T flip flop, master slave flip flops and edge triggered flip flops, state table of a sequential circuit, state diagram, characteristic tables of flip flops, Mealy and Moore machine, flip flop excitation tables, design procedure of clocked sequential circuit, 3-bit binary counter, shift register, ripple counter, RAM

Particulars of course designer:

Name: Dr. Hasin Afzal Ahmed

Contact No.: 8011810533

E-mail id: hasin@gauhati.ac.in

Paper Name: Computer Organization and Architecture

1. Learning Outcome: Student will

- be able to learn about the structure, function and characteristics of computer systems.
- understand the design of the various functional units and components of computers.
- identify the elements of modern instructions sets and their impact on processor design.
- able to learn about the function of each element of a memory hierarchy.
- able to learn about identify and compare different methods for computer I/O.
- Student will able to learn about basics of assembly language.

2. Prerequisite: NIL

3. Semester: 3

4. Course Type: Compulsory

5. Course Level: 200-299

6. Theory credit: 4

7. Practical credit: 0

8. Number of required hours:

- m) Theory: 60 hrs
- n) Practical: 0 hrs
- o) Non Contact: 5 hrs

9. List of books:

- g) M.Morris Mano, *Computer System Architecture*, PHI publication.
- h) Hamachar, Vranesic and Zaky, *Computer Architecture*.
- i) William Stallings, *Computer Organization and Architecture*; Pearson.
- j) Ramesh Gaonkar, *Microprocessor Architecture, Programming, and Applications with the 8085*, 5th Edition.

10. Detailed Syllabus:

UNIT 1: Introduction

(4 Lectures)

Definitions of Computer Organization and Architecture, History of computer architecture, Basic functional blocks of a computer: CPU, memory, Input-output subsystems, Control unit, Types of register- general purpose registers, special purpose registers, index registers.

UNIT 2: Data Representation

(8 Lectures)

Number system, Complements, Representation of signed numbers, Subtraction of unsigned numbers, Fixed-Point representation- Integer representation, Arithmetic addition, Arithmetic subtraction, Overflow, Decimal Fixed-Point representation, Floating-Point representation, Other Binary Codes- Gray Code etc.

UNIT 3: Register Transfer and Micro-operation

(8 Lectures)

Introduction to Register Transfer Language, Register transfer, Bus and Memory transfers, Arithmetic micro-operation- Binary adder, Binary adder-subtractor, Binary incremter, Arithmetic circuit, Logic micro-operation, Shift micro-operation, Arithmetic logic shift unit.

UNIT4: Processing Unit

(10 Lectures)

Instruction codes, Computer registers, General register organization, Register stack, Memory stack, Computer instructions, Data path in a CPU, Operations of a control unit, Hardwired control unit, Micro-programmed control unit, Instruction cycle, Operands, Addressing modes, Instruction format-

Three-address instructions, Two-address instructions, One-address instructions, Zero-address instructions, Data transfer and manipulation- Data transfer instructions, Data manipulation instructions, Arithmetic instructions, Logical and Bit manipulation instructions, Shift instructions, Program Control-Status bit conditions, Conditional branch instructions, Subroutine call and return, Instruction execution cycle, CISC and RISC architectures.

UNIT 5: Memory Organization

(10 Lectures)

Semiconductor memories, Memory cells - SRAM and DRAM cells, Concept of hierarchical memory organization, Interleaved memories, Cache memory unit - Concept of cache memory, Mapping methods, Organization of a cache memory unit, Cache replacement policies, Write policy, Concept of virtual memory.

UNIT 6: I/O Organization

(10 Lectures)

Access of I/O devices, I/O ports, I/O control mechanisms - Program controlled I/O, Interrupt driven I/O, DMA controlled I/O, Interrupts: Types of interrupts, Enabling and disabling interrupts, Handling interrupts.

UNIT 7: Basics of Microprocessor and Assembly Language

(10 Lectures)

Introduction to microprocessors, 8085 Microprocessor and its operation, 8085 instruction sets, Addressing modes in 8085, Classifications of instructions and addressing mode, Assembly language programming basics, Assembling, Executing and debugging the programs, Developing counters and Time delay routines, Interfacing concepts.

Particulars of course designer:

Name: Dr Irani Hazarika

Contact No: 8486965773

Email: queensarathi@gmail.com

Paper Name: Object Oriented Programming through C++

1. **Learning Outcome:** After successful completion of this course, students will be able to:

- Will be able to imagine real-life concepts as objects; derive their properties and functions to operate.
- Develop programs using object- oriented features like data abstraction, polymorphism, inheritance, exception handling.
- Know C++ streams, operators
- Know file handling techniques in C++.

2. **Prerequisite:** NIL

3. **Semester:** 3

4. **Course Type:** Compulsory

5. **Course Level:** 200-399

6. **Theory credit:** 3

7. **Practical credit:** 1

8. **Number of required hours:**

- p) Theory: 45 hrs
- q) Practical: 30 hrs
- r) Non Contact: 5 hrs

9. **List of books:**

- k) M. T. Somashekara, D. S. Guru et-al; *Object-Oriented Programming with C++*, 2nd Edition, PHI,2012.
- l) Bjarne Stroustrup, *The C++ Programming Language*, Special Edition, Pearson Education, 2004.
- m) Deitel&Deitel, *C++ How to program*, Pearson Education Asia, 6th Edition, 2008
- n) Schildt Herbert, *The Complete Reference C++*, Tata McGraw Hill, 4th Edition, 2003.

10. **Detailed Syllabus:**

a) **Theory Content**

UNIT 1: Introduction to object oriented programming

(10 Lectures)

Origins of C++, Basic Concepts of Object Oriented Programming, Benefits of OOP, Applications of OOP, Introduction to C++, Structure of a Simple C++ program, Output operator, Input operator, Cascading of I/O operators, Tokens- keyword, identifiers, constants, strings and operators. Basic data types, User defined data types, Dynamic initialization of variables, Reference variables, Operators in C++, Scope resolution operator & applications, Member dereferencing operators, Memory Management operators, new and delete, Control Structures-simple if, if else, nested if, switch, while do, break and continue statements, Introduction to Functions-Function Prototyping, Call by reference, Return by reference, Inline functions, Default arguments, Constant arguments.

UNIT 2: Classes and objects

(10 Lectures)

Introduction - Defining a class-Class Vs structures, Creating objects, Accessing class members, Defining member functions- Outside the class definition, Inside the class definition, Outside functions as inline, Nesting of member functions, Private member functions, Memory allocation for

objects, Array-Declaring an array-accessing elements of an array, Array of objects, Friendly functions, Constructors and destructors, Basic Concepts of constructors, Default constructor, Parameterized constructor, Multiple constructors in a class, Constructor with default arguments, Dynamic initialization of objects, Copy constructor, Dynamic constructors, Destructors\

UNIT 3: Function and operator overloading (9 Lectures)

Overloading Concepts Function Overloading: Functions with different sets of parameters, default and constant parameters, Rules for overloading operators. Defining operator overloading, Overloading Unary operators, Prefix and Postfix operators overloading, Overloading Binary operators, overloading relational operators, Overloading using friend functions, Overloading subscript operator, Pitfalls of operator overloading, Type conversion-Basic to Class, Class to Basic

UNIT 4: Inheritance (8 Lectures)

Introduction-Defining derived classes, Types of inheritances. Making a private member inheritable, multilevel inheritance, multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base classes, Abstract classes, Constructors in derived classes, nesting of classes, polymorphism-Compile time and Runtime polymorphism, Pointers to objects, this pointer, Pointer to derived classes, Virtual functions, Rules for virtual functions, Pure virtual functions.

UNIT 5: Streams (4 Lectures)

C++ stream classes-put() and get() functions, getline() and write() functions, Overloading << and >> operators, Formatted Console I/O operations, ios class functions-width(), precision(), fill(), setf() and unsetf(), Formatting flags, Manipulators, User defined manipulators.

UNIT 6: Files (4 Lectures)

Introduction-Stream classes for files, Opening files using constructor, Opening files using open(), File modes, Detecting end of file-eof(), Sequential input and output-put() and get()-Reading and writing objects-read() and write()-Random Access files-Manipulating file.

b) Practical / Lab work to be performed

1. Define a class named *triangle* to represent a triangle using the lengths of the three sides. Write a constructor to initialize objects of this class, given the lengths of the sides. Also write member functions to check
 - (a) if a triangle is isosceles
 - (b) if a triangle is equilateralWrite a main function to test your functions.
2. Define a structure *employee* with the following specifications.

empno: integer
ename: 20 characters
basic, *hra*, *da* : float
calculate() : a function to compute net pay as $basic + hra + da$ with float return type.
getdata() : a function to read values for *empno*, *ename*, *basic*, *hra*, *da*.
dispdata() : a function to display all the data on the screen

Write a main program to test the program.
3. Define a class *circle* to represent circles. Add a data member *radius* to store the radius of a circle. Write member functions *area()* and *perimeter()* to compute the area and perimeter of a circle.
4. Define a class *complex* with two data members *real* and *imagto* to represent real and imaginary parts of a complex number. Write member functions

rpart() : to return the real part of a complex number

ipart() : to return the imaginary part of a complex number

add() : to add two complex numbers.

mul() : to multiply two complex numbers.

Write constructors with zero, one and two arguments to initialize objects. (*This is an example of polymorphism.*)

5. Define a class *point* with two data members *xordinate* and *yordinate* to represent all points in the two dimensional plane by storing their x co-ordinate and y co-ordinate values. Write member functions

dist(): to return the distance of the point from the origin.

slope(): to return the slope of the line obtained by joining this point with the origin.

Write constructors with zero, one and two arguments to initialize objects. Also write a friend function to compute the distance between two points.

6. Define a class *string* with the following data members `char *p; int size;` and write member functions to do the following (without using library function) and using dynamic memory allocation.

- Length of the string
- Compare two strings
- Copy one string to another
- Reverse the string

Write suitable constructors and destructors. Also write a copy constructor for the class.

7. For the class *complex* defined in 4 above, overload the `<<`, `>>`, `+` and `*` operators in the usual sense. Also overload the unary `-` operator.
8. For the class *string* defined in 6 above, overload the `<<`, `>>` and `+` operators where `+` is to be used for concatenating two strings.
9. Define a class *time* to store time as hour, minute and second, all being integer values. Write member functions to display time in standard formats. Also overload the `++` and `-` operators to increase and decrease a given time by one second where the minute and hour values will have to be updated whenever necessary.
10. Define a class to store matrices. Write suitable friend functions to add and multiply two matrices.
11. Write a class-based program implementing static members.
12. Define a class *student* with the following specification:

rollno : integer sname : 20 characters

Derive two classes *artst* and *scst*. The class *artst* will represent students belonging to arts stream and the class *scst* will represent students belonging to science stream. The *artst* class will have additional data members *ph*, *hs*, *en* and *as* to store marks obtained by a student in three subjects Philosophy, History, English and Assamese. The class *scst* will have additional data members *ph*, *ch*, *ma* and *ento* to store marks obtained in Physics, Chemistry, Mathematics and English.

Write the following member functions in the classes *artst* and *scst*

ctotal() : a function to calculate the total marks obtained by a student

takedata() : function to accept values of the data members

showdata(): function to display the marks sheet of a student .

13. Define an abstract base class *printer*. Derive three classes *laser-printer*, *line-printer* and *inkjet-printer*. The derived classes will have data members to store the features of that particular printer. Write pure virtual function `display()` in the base class and redefine it in the derived classes.
14. Define an abstract base class *figure* and add to it pure virtual functions. Derive three classes *circle*, *rectangle* and *triangle* from it. A circle is to be represented by its radius, rectangle by its length and breadth and triangle by the lengths of its sides. Write a main function and write necessary statements to achieve run time polymorphism.\
15. Write an interactive program to compute square root of a number. The input value must be tested for validity. If it is negative, the user defined function *my_sqrt()* should raise an exception.
16. Define a class *rational* to store rational numbers as a pair of integers, representing the numerator and denominator. Write a member function for setting the values of the numerator and denominator. This function should raise an exception if attempt is made to set a zero value as the denominator and in such cases it should be set to 1.
17. Write a class template for storing an array of elements. Overload the << and >> operators. Write a member function to sort the array in descending order.
18. Write a class template for representing a singly linked list. Write functions for inserting, deleting, searching and for displaying a linked list. Write a main function to test it on a linked-list of integers and characters.

Particulars of course designer:

Name: Prof. Anjana Kakoti Mahanta

Contact No.: 9864425716

E-mail id : anjana@gauhati.ac.in

Paper Name: Operating System

1. **Learning Outcome:** After completing this course, students will have understanding of the internal structure and usage of various components related to an operating system.
2. **Prerequisite:** NIL
3. **Semester:** 3
4. **Course Type:** Compulsory
5. **Course Level:** 200-399
6. **Theory credit:** 3
7. **Practical credit:** 1
8. **Number of required hours:**
 - s) Theory: 45 hrs
 - t) Practical: 30 hrs
 - u) Non Contact: 5 hrs
9. **List of books:**
 - o) Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Wiley
 - p) Modern Operating Systems, Andrew S. Tanenbaum, Prentice-Hall Of India Pvt. Limited

10. Detailed Syllabus:

Unit I: Introduction

7 hrs

Application vs system software, operating system as system software, operating structure structure, types of operating systems: batch operating system, multiprogramming operating system, multi tasking operating system, distributed operating system, real time operating system, multi user operating system, major functions of operating system: Process Management, Process Synchronization, Memory Management, CPU Scheduling, File Management, I/O Management, Security, virtualization, cloud computing, open source operating system, history of operating system, the shell, system call, system boot

Unit II: Process and threads

10 hrs

Process, process states: new, running, waiting, ready and terminated, Process Control Block (PCB), information stored in PCB, scheduling queue: job queue, ready queue and device queue, schedulers: long term schedulers, medium term scheduler and long term scheduler, swapping, degree of multiprogramming, I/O-bound and CPU-bound processes, context switching, inter-process communication: shared memory systems and message passing systems, socket, remote procedure call, threads, user threads, kernel threads, multi threading models: Many-to-One Model, One-to-One Model, Many-to-Many Model, CPU scheduling, Scheduling Criteria, scheduling algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling

Unit III: Process synchronization

8 hrs

Race condition, critical section problem, Peterson's algorithm, Bakery algorithm, synchronization hardware: locking, synchronization software tools: mutex lock, semaphore (counting and binary), semaphore implementation, classic synchronization problems: bounded buffer problem, the readers-writers Problem, the dining-philosophers problem, monitor, synchronization in windows, synchronization in linux

Unit IV: Deadlock**10 hrs**

Deadlock, operations of a process performs while using a resource: Request. Use and Release, physical and logical resources, Necessary conditions: mutual exclusion, hold & wait, no preemption and circular wait, resource allocation graph, deadlock prevention: definition, preventing mutual exclusion, preventing hold & wait, preventing no preemption and preventing circular wait, deadlock avoidance: definition, safe state, safe sequence, resource allocation graph based algorithm and Banker's algorithm, deadlock detection: definition, wait-for graph, algorithm to detect deadlock for single instance resources, algorithm to detect deadlock for multiple instance resources and recovery from deadlock: process termination and resource preemption

Unit V: Memory Management**10 hrs**

Memory hierarchy, base register, limit register, address binding, logical and physical address spaces, memory management unit, relocation register, swapping, contiguous memory allocation: definition, memory protection, fixed partition scheme, variable partition scheme, first-fit, best-fit & worst-fit allocation strategies, non-contiguous memory allocation: simple paging and simple segmentation, internal and external fragmentation, TLB, virtual memory, demand paging, page fault, locality of reference principle, performance of demand paging, page replacement algorithms: FIFO, Optimal and LRU, allocation of frames: equal allocation and proportional allocation, global and local page replacement algorithms, thrashing

(b) Practical

- **Basic linux commands:** pwd, ls, cd, mkdir, rmdir, rm, touch, man, cp, mv, locate, head, tail
(2 Classes/4 hrs)
- **Advanced commands:** echo, cat, sudo, df, tar, apt-get, chmod, hostname, useradd, passwd, groupadd, grep, sed, uniq, wc, od, gzip, gunzip, find, date, cal, clear, top, ps, kill (3 Classes/6 hrs)
- **Shell scripting in linux:** shell, types of shell, shell script, echo command, shell variables, special variables (\$\$, \$0, \$n, \$#, \$?, \$!), array, assignment operator (=), equality operator (==), not equality operator (!=), arithmetic operators (+, -, *, /, %), comparison operators (-eq, -neq, -gt, -lt, -ge, -le), logical operators (!, -o, -a), if...else statement, case...esac statement, while loop, for loop, break statement, continue statement, shell functions (7 Classes/14 hrs)
- **Using system calls in C program in linux:** fork(), exec(), exit(), getpid(), mkdir(), rmdir() etc. (3 Classes/6 hrs)

Particulars of course designer:**Name:** Dr. Hasin Afzal Ahmed**Contact No.:** 8011810533**E-mail id:** hasin@gauhati.ac.in

Paper Name: Automata Theory and Languages

1. Learning Outcome: After completing this course, students

- Understand the Mathematical model of a finite state machine. Know deterministic and non-deterministic versions of Finite automata.
- Grasp the mathematical concepts of languages and grammar.
- Know Pushdown Automata and the associated grammar/language.
- Know the properties of Regular languages and Context free languages.

2. Prerequisites: NIL

3. Semester: 4

4. Course Type: Compulsory

5. Course Level: 200-299

6. Theory Credit: 4

7. Practical Credit: 0

8. No of Hours:

- a) Theory: 60 hrs
- b) Practical: 0 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) *An introduction to Formal Languages and Automata*, Peter Linz, Narosa.
- b) *Introduction to Automata Theory, Languages and Computation*, Hopcroft, Motwani and Ullman, Pearson.
- c) *Theory of Computer Science (Automata, Languages and Computation)*, K. L. P. Mishra, N. Chandrasekaran; P. H.I.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Finite Automata

(10 Lectures)

DFA, NFA, NFA with empty-moves, Equivalence of DFA and NFA, Reduction of the number of states in finite automata.

UNIT 2: Regular Languages and Regular Grammar

(12 Lectures)

Concept of languages and grammar, Regular expressions, Connection between regular expressions and regular languages, Regular grammars, Right and Left-Linear Grammars, Equivalence between Regular languages and Regular grammars.

UNIT 3: Properties of Regular Languages

(13 Lectures)

Closure under simple set operations- union, intersection, concatenation, complementation and star closure, Decision algorithms for emptiness, finiteness and infiniteness, equality, Proof of non-regularity using Pigeonhole principle and using pumping lemma for regular languages.

UNIT 4: Context Free languages

(15 Lectures)

Context-free grammars, leftmost and rightmost derivations, derivation trees, Parsing and Ambiguity in grammars and languages, Simplification of Context free Grammars- removing useless productions, empty-productions and unit-productions. Normal forms- Chomsky and Greibach normal forms, Pumping Lemma for CFL, Using Pumping Lemma to show that certain languages are not Context free

UNIT 5: Pushdown Automata**(10 Lectures)**

Definition and language accepted (acceptance by empty stack and final state and their equivalence), Pushdown Automata and Context free languages. Deterministic PDA and Deterministic Context free Languages.

Particulars of course designer:

Name: Prof. Anjana Kakoti Mahanta

Contact No.: 9864425716

E-mail id : anjana@gauhati.ac.in

Paper Name: Database Management System

1. Learning Outcome:

On successful completion of this course, the student should be able to:

- Learn database concepts and its architectural components.
- Describe different data models used for designing a database.
- To create a database using relational models and entity relationships concepts
- Normalize a database into various normal forms
- Design SQL queries to handle a relational database.

2. Prerequisite: NIL

3. Semester: 4

4. Course Type: Compulsory

5. Course Level: 200-299

6. Theory Credit: 3

7. Practical Credit: 1

8. Number of required hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of reference books:

- a) Dr. Satinder Bal Gupta and Aditya Mittal, *Introduction to Database Management System*, University Science Press
- b) A. Silberschatz, H.F. Korth, S. Sudarshan, *Database System Concepts*, McGraw Hill
- c) R. Elmasri, S.B. Navathe, *Fundamentals of Database Systems*, Pearson Education
- d) Dr. Rajive Chopra, *Database Management System (DBMS): A Practical Approach*, S. Chand Publication

10. Detailed Syllabus:

UNIT-1: Introduction to Database Management Systems

(5 Lectures)

Basic Definition and Concepts: *Data, Information, Meta Data, Data Dictionary, Database, Fields, Records and Files*. Definition of Database Management System (DBMS), Primary Functions of DBMS, Traditional File approach, Traditional file approach versus database management system approach, Disadvantages of Traditional File System, Need of a DBMS, Components of a DBMS, Advantages of DBMS, Disadvantages of Database Systems, Various uses of database System Applications, Database Users: *End users or naive users, Online users, Application Programmers, Database Administrator(DBA)*, Responsibilities of DBA.

UNIT 2: Database Management System Architecture

(6 Lectures)

Definition of *Schemas, sub-schema and Instances*. Data Independence: *Physical Data Independence and Logical data Independence*. Three-tier architecture of DBMS, Advantages of three-level Architecture, basic concept of data model, Characteristics of Data Models, Types of Data models: *Record Based Data Models, Object Based Data Model and Physical Data Models*. Relational Data Model, Types of database Systems: *Single-user database systems, Multiuser database systems, Centralized database systems, Distributed database systems and Client/Server database systems*.

UNIT 3: E-R Modeling**(8 Lectures)**

Basic Concepts: *Entity, Attributes, Entity Sets, Domain*. Types of attributes: *Simple and Composite Attributes, Single Valued and Multi-valued Attributes, Derived Attributes and Stored Attributes*. Types Of Entity Sets: *Strong Entity Sets* and *Weak Entity Sets*. Concept of Relationship and Relationship sets, Types of Relationship: *One-to-One, One-to-Many, Many-to-One and Many-to Many*, Various Symbols used in ER Diagram, Mapping constraints: *Mapping Cardinalities (Cardinality Ratios)* and *Participation Constraints*. Definition of Key, Types of Keys: *Super Key, Candidate Key, Primary Key, Alternate Key and Foreign Key*. Symbols used in E-R diagrams, Conversion of an ER and Diagram in to Relational Tables

UNIT4: Relational Model and Relational Algebra**(7 Lectures)**

Definition of Relation, Data Structure of Relational Database: *Relation, Tuples, Attributes Domain, Degree and Cardinality*. Integrity Constraints, Domain Constraints, Key Constraints, Advantages and Disadvantages of Relational Model, Relational, Definition of Relational algebra, Operations in Relational Algebra: *Selection, Projection, Division, Rename, Union, Intersection, Set Difference, Natural-join operation, Outer join, Inner Join, Cartesian Product and Assignment operation*. Aggregate Functions and Operations: *Average, Maximum, Minimum, Sum and Count*.

UNIT 5: Functional Dependency and Normalization**(8 Lectures)**

Definition of Functional Dependency, Armstrong's Axioms in Functional Dependency, Types of Functional Dependency: *Partial Dependency, Full Functional Dependency, Transitive and Non-transitive Functional Dependency*, Armstrong's Axiom, Closure of a set of Functional Dependency, Closure of an Attribute, Definition of Canonical Cover, Algorithm to find the canonical cover of a FD set, Anomalies in relational database: *Insertion, Deletion and Update anomalies*, Concepts of Normalization, Benefits of Normalization, Types of Normal Forms: *First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF) and Boyce-Codd Normal Form (BCNF)*

UNIT 6: Transaction and Concurrency Control**(4 Lectures)**

Definition of Transaction, ACID Properties of transaction, Transaction States, Definition of Concurrency Control, Need of Concurrency Control, The Lost Update Problem, The Uncommitted Dependency Problem, The Inconsistent Analysis Problem, Serializability: *View Serializability and Conflict Serializability*

UNIT 7: SQL Queries**(7 Lectures)**

Database Languages (Data Definition Languages, Data Manipulation Languages), Characteristics of SQL, Basic data types in SQL, Data-definition language (DDL) commands: *Create Database, Create Table, Drop Table, Alter Table*. SQL Constraints: *Primary Key, Foreign Key, Not Null, Unique, Check, Default*,. Data Manipulation Language (DML) commands: *Insert Into, Delete, Select, Update*. SQL clauses: *Where, Order By, Having, Group By* and *Like*. SQL join operations: *Inner Join, Left Outer Join, Right Outer Join* and *Full Join*. SQL aggregate functions: *sum(), count(), max(), min()* and *avg()*

Lab Contents: (30 hrs)

Practical / Lab work to be performed:

- Implementation of SQL DDL statements in MySQL DBMS: CREATE DATABASE, CREATE TABLE, ALTER TABLE, RENAME, DROP DATABASE/TABLE
- Use of SQL DML statements in MySQL DBMS: INSERT, SELECT, UPDATE, DELETE SQL commands
- Implementing following constraints in MySQL DBMS: PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE and DEFAULT
- Handling following SQL clauses in MySQL DBMS: WHERE, GROUP BY, ORDER BY, HAVING, IN, BETWEEN, LIKE
- Working with following aggregate functions in MySQL DBMS: COUNT, AVG, MAX, MIN and SUM
- Working with transaction processing command in MySQL DBMS: START TRANSACTION, COMMIT and ROLLBACK Statements, SET autocommit

Particulars of course designer:

Name : Dwipen Laskar

Contact No : +916000795681

Email-id : laskardwipen@gauhati.ac.in

Paper Name: Design and Analysis of Algorithms

1. Learning Outcome:

After successful completion of this course, students will:

- know how to analyze algorithms.
- learn the different algorithm design techniques.
- be acquainted with the advanced sorting and searching algorithms and their complexities.
- know graph representation techniques together with traversal algorithms.
- know why tree balancing is required and how to achieve this.

2. Prerequisites: NIL

3. Semester: 4

4. Course Type: Compulsory

5. Course Level: 200-299

6. Theory Credit: 4

7. Practical Credit: 0

8. No of Hours:

- a) Theory: 60 hrs
- b) Practical: 0 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) *Introduction to Algorithms*, Cormen. T. H., Leiserson C. E. and Rivest. R. L., 3rd edition (2010)Tata-McgrawHill Publishers.
- b) *Fundamentals of Computer Algorithms*; Horowitz and Sahani; (2nd Edition), Galgotia.
- c) *Design and Analysis of Computer Algorithms*; Aho.A, Hopcroft J.E. and Ullman J.D.; (2011), PearsonEducation.
- d) *Introduction to the Design and Analysis of Algorithms*, Levitin, 3/e 2017, Pearson Education.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction

(6 Hours)

Analysis of Algorithms – worst case and average case analysis; Time and space complexity of algorithms; Asymptotic notations O and θ . Proving correctness of algorithms.

UNIT 2: Algorithm Design Techniques

(10 Hours)

Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms. Applications of these techniques in problems like sorting, searching, matrix multiplication, LCS (Longest Common Sequence) problem, Knap-sack problem.

UNIT 3: Sorting and Searching Techniques

(20 Hours)

Elementary sorting techniques–Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Counting Sort, Searching Techniques, Medians & Order Statistics, complexity analysis of all the techniques.

UNIT 4: Balanced Trees

(9 Hours)

Tree balancing, Height of a Red-Black tree, Rotations - Left Rotations, Right Rotations, Insertion and Deletion in Red-Black trees.

UNIT 5: Graph Algorithms**(9 Hours)**

Representations of Graphs; Adjacency Matrix and Adjacency Lists. Simple operations like computing degree, indegree, outdegree of vertices using the representation techniques and computing work done in all cases. Graph traversal algorithms–Breadth First Search, Depth First Search and their Applications.

UNIT 6: String Processing**(6 Hours)**

String Matching, KMP Technique.

Particulars of course designer:

Name: Prof. Anjana Kakoti Mahanta

Contact No.: 9864425716

E-mail id : anjana@gauhati.ac.in

Paper Name: Python Programming

1. Learning Outcome: After completing this course, students

- Know about fundamentals of Python Programming and Problem Solving.

2. Prerequisites: NIL

3. Semester: 4

4. Course Type: Compulsory

5. Course Level: 200-299

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- d) *Core Python Programming*, R. Nageswara Rao, Dreamtech Press.
- e) *Python: The Complete Reference*, Martin C. Brown, McGraw Hill Education.
- f) <http://docs.python.org/3/tutorial/index.html>

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Introduction to Python Programming (8 hrs)

Introduction, Installation of Python Interpreter, Python Shell, Code Indentation, Identifiers and Keywords, Literals, Strings, Operators (Arithmetic, Relational, Logical, Assignment, Ternary, Bitwise, Increment and Decrement Operators), Input and output statements, Output Formatting.

Unit 2: Control Statements and Functions (8 hrs)

Branching, Looping, Conditional Statement, Exit Functions, Break, Continue, Pass, Defining Functions, Default Arguments. Scope of Functions, Function Documentation, Lambda Functions & Map.

Unit 3: Python Data Structures (6 hrs)

List (List, Nested List, List as Matrix), Tuple, Set, Dictionary.

Unit 4: Exception Handling (4 hrs)

Errors, Exception Handling with try, Multiple Exception Handling, Writing own Exception.

Unit 5: File Handling (6 hrs)

Understanding read function, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Programming using file operations, Reading config files, Writing log files in python.

Unit 6: OOP in Python

Creating Classes in Python, Instance Methods, Inheritance, Polymorphism, Exception Classes and Custom Exceptions.

Unit 7: Introduction to Libraries in Python (6 hrs)

NumPy, Matplotlib, OpenCV, Tkinter.

Unit 8: Python SQL Database Access**(7 hrs)**

Introduction to database driven program, Database Connection, Database Operations: INSERT, READ, UPDATE, DELETE, COMMIT AND ROLLBACK.

(b) Practical

- Introduction to Python console, operators, input and output statements.
- Python control statements and functions
- Data Structures in python
- Exception Handling
- File Handling
- Object Oriented Python programming
- Introduction to libraries (NumPy, Matplotlib, OpenCV)
- Python SQL Database Connection and database operations

Particulars of course designer:

Name: Dr. Sanjib Kr Kalita

Contact No.: 8812051150

E-mail id: sanjib959@gauhati.ac.in

Paper Name: Computer Networks

1. Learning Outcome: After completing this course, students

- Student will able to learn about the general principles of data communication.
- Student will able to learn about how computer networks are organized with the concept of layered approach.
- Student will able to learn about how signals are used to transfer data between nodes.
- Student will able to learn about how packets in the Internet are delivered.
- Student will able to learn about how routing protocols work.
- Student will able to learn about functions of transport layer
- Student will able to learn about functions of application layer

2. Prerequisites: NIL

3. Semester: 5

4. Course Type: Compulsory

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- g) B. A. Forouzan: *Data Communications and Networking*, Fourth edition, THM, 2007.
- h) A. S. Tanenbaum: *Computer Networks*, Fourth edition, PHI , 2002.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction to Computer Networks

(5

Lectures)

Data communication system and its components, Definition of network, Types of network, Network topologies, Network protocol, Layered network architecture, Overview of OSI reference model, Overview of TCP/IP protocol suite.

UNIT 2: Physical Layer Communication

(10 Lectures)

Analog and digital signal, Definition of bandwidth, Maximum data rate of a channel, Line encoding schemes, Transmission modes, Modulation techniques, Multiplexing techniques- FDM and TDM, Transmission media-Guided and Unguided, Switching techniques- Circuit switching, Packet switching, Connectionless datagram switching, Connection-oriented virtual circuit switching.

UNIT 3: Data Link Layer Functions and Protocol

(10 Lectures)

Definition of Framing, Framing methods, Error detection techniques, Error correction techniques, Flow control mechanisms- Simplex protocol, Stop and Wait ARQ, Go-Back-N ARQ, Point to Point protocol.

UNIT 4: Multiple Access Protocol and Networks

(5 Lectures)

Basics of ALOHA protocols, Basics of CSMA/CD protocols, Ethernet LANS, Connecting LAN and back-bone networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways

UNIT 5: Networks Layer Functions and Protocols**(8 Lectures)**

Connection oriented vs Connectionless services, Definition of Routing, Routing algorithms, IP protocol, IP addresses, ARP, RARP

UNIT 6: Transport Layer Functions and Protocols**(4 Lectures)**

Transport services, TCP vs UDP protocol, TCP connection establishment- Three way handshakes, TCP connection release

UNIT 7: Overview of Application Layer Protocols**(3 Lectures)**

Overview of DNS, Overview of WWW, URL, Email architecture, HTTP protocol

B. Practical / Lab work to be performed**(15 Practical Classes/30 hrs)**

- Implement the data link layer framing methods such as Bit Stuffing.
- Study of different types of Network cables.
- Study of network IP.
- Connect the computers in Local Area Network.
- Study of basic network command and Network configuration commands.
- Configure a Network topology using packet tracer software.
- Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
- Simulate and implement Stop and Wait protocol for noisy channel.
- Simulate and implement Go-Back-N sliding window protocol.
- Simulate and implement Selective Repeat sliding window protocol.
- Simulate and implement Dijkstra Algorithm for shortest path routing.
- Simulate and implement Distance vector routing algorithm

Particulars of Course Designer:

Name: Dr Irani Hazarika

Contact No: 8486965773

Email: queensarathi@gmail.com

Paper Name: Java Programming

1. Learning Outcome: After completing this course, students will be

- Familiar with the core concepts of java programming and classes of swing package.

2. Prerequisites: NIL

3. Semester: 5

4. Course Type: Compulsory

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) *Java: The Complete Reference*, Herbert Schildt, McGrawHill
- b) *Java How to Program*, Paul Deitel, Harvey Deitel, Pearson

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit I: Introduction

(3 hrs)

High level language, compiled and interpreted languages, history of java programming language, compilation of java code, bytecode, java interpreter, javac and java command, path environmental variable, Java IDE, features of java programming language: simple, object oriented, robust, architecture neutral and interpreted

Unit II: Data types, operators and control statements

(12 hrs)

Java as strongly typed language, primitive data types, integer data types: byte, short, int and long, floating point data types: float and double, character data type, boolean data type, literals: integer literals, floating-point literals, boolean literals, character literals and string literals, declaring a variable, dynamic Initialization, the scope and lifetime of variables, type-casting in java, one dimensional array, multi dimensional array, arithmetic operators: the basic arithmetic operators, the modulus operator, arithmetic compound assignment operators, increment operator and decrement operator, bitwise operators, relational operators, short circuit logical operator, the assignment operator, branching statements: if-else and switch-case statements, looping statements: while, do-while, for and for-each statements, jump statements: break and continue

Unit III: Object oriented features of java

(10 hrs)

Defining a class, member variable and member methods, access specifiers: default, private and public, declaring objects, assigning object reference variables, constructors, parameterized constructors, the this keyword, garbage collection, the finalize() method, overloading methods, overloading constructor, static keyword, final keyword, command line arguments in java, inheritance, super class and sub class, protected access specifier, super keyword, constructor call in multilevel inheritance, method overriding, dynamic method dispatch, abstract class, interfaces, type wrappers

Unit IV: String handling and packages

(5 hrs)

String class, String constructors, String length, special string operations: string literals, string concatenation, string concatenation with other data types, string conversion and toString(), character extraction: charAt(), getChars(), string Comparison: equals() and equalsIgnoreCase(), regionMatches(), startsWith() and endsWith(), equals() Versus ==, compareTo(), searching strings, data conversion using valueOf(), StringBuffer, StringBuffer constructors, length() and capacity(),

ensureCapacity(), setLength(), charAt() and setCharAt(), getChars(), package, defining a package, CLASSPATH, importing packages

Unit V: Exception handling and I/O

(5 hrs)

Exception-handling, exception types, uncaught exceptions, try and catch block, multiple catch blocks, nested try statements, throw, throws, finally, java's built-in exceptions, creating own exception classes, java I/O classes, reading console input, writing console output, reading and writing files

Unit VI: Swing package and database connectivity

(10 hrs)

Swing package, simple GUI-Based Input/Output with JOptionPane, JFrame, JLabel, JTextField, JButton, handling event in a JFrame object, layout managers: BorderLayout, FlowLayout, GridLayout, CardLayout, GridBagLayout, JToggleButton, JCheckBox, JRadioButton, JList, JComboBox, JDBC, JDBC driver, connectivity steps, connectivity with MySQL, DriverManager class, Connection class, Statement class, ResultSet class, PreparedStatement class

(b) Practical

- Java programs to demonstrate the use of data types and operators
- Java input through Scanner class and JOptionPane class
- Java programs to demonstrate the use of control statements.
- Java programs to demonstrate the use of classes, objects, visibility modes, constructors and destructor.
- Java programs to demonstrate the use of inheritance and polymorphism.
- Java programs to demonstrate the use of polymorphism.
- Java programs to handle strings, Java programs implementing exception handling.
- Demonstrating the use and creation of packages in java.
- Java program with JFrame, JTextField and JButton with event handling
- Using JLabel, JTextArea and JPasswordField in java with event handling
- Working with layout managers in JFrame
- Using JCheckBox, JRadioButton and JComboBox in a JFrame
- Connecting JFrame components to a DBMS

Particulars of course designer:

Name: Dr. Hasin Afzal Ahmed

Contact No.: 8011810533

E-mail id: hasin@gauhati.ac.in

Paper Name: Software Engineering

1. Learning Outcome: On successful completion of this course, the student should be able to:

- Determine the primary problems that impact all software development processes.
- Choose relevant software development processes models, methodologies, and strategies for managing a specific software development process, and justify the choices
- Implement different software estimation metrics such as cost, effort size, staffing etc.
- Describe various software design approaches and various coding and testing strategies used in software engineering principles
- Know about software reliability and how to calculate software maintenance cost.

2. Prerequisites: NIL

3. Semester: 5

4. Course Type: Compulsory

5. Course Level: 300-399

6. Theory Credit: 4

7. Practical Credit: 0

8. No of Hours:

- a) Theory: 60 hrs
- b) Practical: 0 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- c) Rajib Mall: *Fundamentals of Software Engineering*; PHI Learning Pvt. Ltd.
- d) Roger S. Pressman: *Software Engineering: A practitioner's Approach*; McGraw Hill.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Introduction

(4 Lectures)

Definition of Software Engineering, differentiation between Computer Science, Software Engineering and System Engineering, Program V/s software product, Exploratory style and modern style of software development, need of software engineering, characteristics of good software product

Unit 2: Software Development Life Cycle models

(7

Lectures)

Definition of software development Life cycle (SDLC) models, Various life cycle modes: Classical Waterfall model, Iterative Waterfall model, Prototyping model, Evolutionary (Incremental) model, Spiral model, Agile Model, Agile V/s traditional SDLC Models, SCRUM model, Advantages and disadvantages of each of these SDLC models.

Unit 3: Requirement Analysis and Specification

(7 Lectures)

What is Requirement Analysis and Gathering, Concept and Importance of Feasibility Study in Software design, Types of Feasibility: *Technical*, *Economical* and *Operational* feasibility, Software Requirement Specification (SRS) document, Components of an SRS (Software Requirement Specification): Functional and Non-Functional Component, Properties of a good SRS, Different users of SRS, Techniques to represent Complex Logic in SRS: Decision Tree and Decision Table.

Unit 4: Software Project Management**(15 Lectures)**

Basic idea of Software Project Management, Job Responsibilities of a Software Project Manager, Need of SPMP (Software Project Management Plan) document, Contents of SPMP, Need of Software documentation, Internal and External documentation, Software size estimation using Lines of Code (LOC), Merits and Demerits of LOC metric, Function Point Metric, 3D Function Point metrics, Project Estimation Techniques: *Empirical estimation* and *Heuristics estimation* techniques. Empirical estimation techniques: *Delphi Cost Estimation* and *Delphi Cost Estimation*. Heuristic Estimation Techniques: *Basic COCOMO model* and *Intermediate COCOMO model*. Project Scheduling: *Work break down structure*, *Activity Networks* and *Critical Path Method*. Project Team structure: *Chief Programmer team* and *Democratic team* structure.

Unit 5: Software Design principles and Methodology**(12 Lectures)**

Top down and bottom up approach, External Design, Architectural Design and Detailed design, Concept of Cohesion in software design, Classification of Cohesions, Basic concept of Coupling, Classification of Couplings, Introduction to software Analysis and Software Design (SA/SD), Introduction to Data Flow Diagram, Symbols used in DFD, Context Diagram in DFD, Advantages and Disadvantages of DFDs., Balanced DFD, Structured Design: *Transaction Analysis* and *Transform Analysis*. Need of Object Oriented Design and Analysis, UML (Unified Modeling Language), different views of UML, Various UML Diagrams: *Use Case diagram*, *Class Diagram*, *Object Diagram*, *Sequence Diagram* and *Collaboration diagram*.

Unit 6: Coding and Testing**(9 Lectures)**

Goals of coding, Code Review techniques: Code Walkthrough, Code Inspection, Definition of Test cases, test suits, negative testing and positive testing. Different levels of software testing: *unit testing*, *Integration Testing*, *System Testing* and *acceptance testing*. Differentiation between Verification and Validation, Black box testing approaches: *Equivalent Class Partitioning* and *Boundary Value Analysis*, White Box testing approaches: *Statement Coverage*, *Branch Coverage*, *Condition Coverage* and *Path Coverage*. Approach, McCabe's Cyclomatic Complexity, Basic idea of various system testing approaches: *Smoke testing*, *Stress testing*, *Volume testing* and *Compatibility testing*

Unit 7: Software Reliability and Maintenance**(6 Lectures)**

What is reliability? Reliability metrics of Software Products: ROCOF, MTTF, MTTR, MTBF, POFOD and availability. ISO 9000 Certification, need of ISO Certification, How to get ISO 9000 certification, Definition of Software Maintenance, Types of Software maintenance: *Corrective*, *Adaptive* and *Perfective* maintenance, Estimation of Software Maintenance Cost.

Particulars of course designer:

Name : Dwipen Laskar

Contact No : +916000795681

Email-id : laskardwipen@gauhati.ac.in

Paper Name: Web Technologies

1. Learning Outcome: At the end of the course, students will be able to:

- Understand the basic concept of web applications and web services.
- Design basic well-structured web page using HTML and CSS
- Develop the ability to implement interactive elements and dynamic content using basic JavaScript
- Develop a foundational understanding of server-side scripting using PHP

2. Prerequisites: NIL

3. Semester: 5

4. Course Type: Compulsory

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) Jackson J.C. (2007). *Web Technologies: A Computer Science Perspective*. Pearson.
- b) Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
- c) Robbins, J. N. (2018). *A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media.
- d) Robbins, J. N. (2018). *Learning Web Design: A Beginner's Guide*. O'Reilly Media.
- e) Haverbeke, M. (2018). *Eloquent JavaScript*. No Starch Press.
- f) Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development (5th ed.)*. Addison-Wesley Professional.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Introduction to Web Technologies

(8 Lectures)

Concepts of the Internet and the World Wide Web (WWW), Overview of web browsers and their functionalities. Client-Server Architecture in Web Applications. Communication Protocols – HTTP, HTTPS, FTP. Working of DNS. Brief concepts of port, URL, cache and cookies. Web Content Accessibility Guidelines. Privacy concerns and data protection regulations, GDPR. Introduction to Web Hosting and control panels.

Unit 2: Front End Development using HTML

(10 Lectures)

Website and Webpage. Basic concept of Markup Language. Introduction to HTML. Basic HTML structure. Text formatting Tags – headings, paragraph, line break, horizontal rule. Link and Navigation – anchor tags. Lists - ordered, unordered, definition list. Image and multimedia tags. Tables in HTML. Forms and Input types – text, email, password, radio, select, checkbox, textarea, date, url, submit, button. Semantic HTML. Sectioning elements – header, nav, main, section, article, aside, footer.

Unit 3: Front End Design using CSS

(9 Lectures)

Introduction to CSS. CSS syntax and rule structure. Inline, Internal and External CSS. CSS selectors – element, class, ID, attribute. Combinators – descendant, child, adjacent sibling, general sibling. Understanding the CSS Box Model – content, padding, border, margin. CSS colours and backgrounds – background-color, background-image, background-repeat. CSS typography – font properties, text properties.

Unit 4: Client-Side Scripting with JavaScript

(10 Lectures)

JavaScript as a high-level interpreted language. JavaScript code execution in web browsers – JavaScript execution context. JavaScript syntax and datatypes. JavaScript variables – var, let, const. Assignment and scope of JavaScript variables. Operators in JavaScript – arithmetic, comparison, logical, assignment. Conditional Statements. Looping Structures. Function declaration and Invocation in JavaScript. Introduction to the Document Object Model. Accessing HTML elements in DOM – by id, by tag name, by class name, query selectors. Manipulating DOM elements – create, add, append, remove. InnerText vs InnerHTML. Manipulating CSS styles using DOM. Event handling and delegation with the DOM using JavaScript. Client-side form validation using JavaScript. Handling form validation and processing data.

Unit 5: Server-Side Programming with PHP

(8 Lectures)

Introduction to PHP and role in Web development. PHP syntax and variables. Basic PHP functions – Built-in PHP functions, string manipulation functions, mathematical functions, date and time functions. PHP forms and form handling. Form submission methods – GET and POST. Handling form data with PHP. Uploading files with PHP. Introduction to the tech-stack. Role of Apache, PHP, MySQL etc. Introduction to Databases and SQL. Connecting to databases with PHP. Executing SQL queries with PHP. Retrieving, inserting, updating and deleting data from databases using PHP.

B. List of Practical

(This is a suggestive list only. Questions need not be restricted to this list.)

1. Create a basic HTML webpage structure with a heading, paragraph, and an image.
2. Build a navigation menu using an unordered list () with clickable links.
3. Implement a form with input fields for name, email, and a submit button.
4. Create a table with multiple rows and columns to display tabular data.
5. Design an image gallery using HTML and CSS with proper padding and border.
6. Embed a YouTube video on a webpage using the <iframe> tag.
7. Implement an ordered list () to display a step-by-step tutorial or instructions.
8. Create a dropdown select menu (<select>) with multiple options.
9. Use HTML5 semantic tags (such as <header>, <nav>, <section>, <article>, <footer>) to structure and organize content on a webpage.
10. Build a registration form with fields for name, email, password, date of birth, address and other such fields with a submit button. Include appropriate input types, labels and placeholders.
11. Style a heading element with a custom font, colour and background.
12. Apply different background colors to alternate rows in a table.
13. Implement a hover effect on a button that changes its background colour or adds a solid border.

14. Style a form input field with custom border, padding, and background color.
15. Implement a CSS tooltip that displays additional information when hovering over an element.
16. Build a simple JavaScript calculator that can perform basic arithmetic operations.
17. Create a button that, when clicked, appends a new paragraph element with a specific text content to an existing div element.
18. Implement a function that changes the innerText of a paragraph element to display a random number between 1 and 10 every time a button is clicked.
19. Build a form with input fields for name and email. When the form is submitted, use innerHTML to display a confirmation message with the entered name and email on the webpage.
20. Build a form with input fields for email, password and confirm password. When the form is submitted, use an alert to display a success message if the password and confirm password values matches, otherwise show an error alert. Use JavaScript for the validation.
21. Create a list of items. Add a click event listener to each item so that when clicked, the background color of the clicked item changes.
22. Write a PHP script to display the current date and time on a webpage.
23. Write a PHP script to connect to a MySQL database and fetch data from a table.
24. Create a registration form with fields for username, email, and password. Implement server-side validation to check for duplicate usernames or invalid email formats. Store the user registration data in a MySQL database. Provide feedback to the user upon successful registration or display appropriate error messages.
25. Design a webpage that displays a list of notices retrieved from a MySQL database. Implement functionality to add new notices to the database using a form. Allow users to view and delete individual notices. Apply appropriate styling to the notices and ensure proper validation and sanitization of user input.

Particulars of Course Designer:

Name : Risheraj Baruah

Contact No. : +91 8486942427

Email id : rishirajbaruah@gauhati.ac.in

Paper Name: Advanced Web Programming

1. Learning Outcome: At the end of the course, students will be able to:

- a) Design basic well-structured web page using HTML and CSS
- b) Develop the ability to implement interactive elements and dynamic content using basic JavaScript
- c) Develop a foundational understanding of server-side scripting using PHP
- d) Create a CRUD web application using HTML, CSS, JavaScript, PHP and MySQL.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons.
- b) Robbins, J. N. (2018). Learning Web Design: A Beginner's Guide. O'Reilly Media.
- c) Nixon, R. (2014). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (4th ed.). O'Reilly Media.
- d) Duckett, J. (2014). JavaScript and JQuery: Interactive Front-End Web Development. John Wiley & Sons.
- e) Haverbeke, M. (2018). Eloquent JavaScript. No Starch Press.
- f) Welling, L., & Thomson, L. (2016). PHP and MySQL Web Development (5th ed.). Addison-Wesley Professional.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit 1: Advanced HTML

(6 Lectures)

Review of basic HTML tags and their usage. Working with forms – validation using HTML5 attributes. HTML5 Semantic Elements – header, nav, section, article, aside, footer. Applying proper semantic markup for improved SEO. Multimedia integration. Embedding images with different attributes. Adding video and audio. Meta information and Document Structure – metadata, viewport settings.

Unit 2: Advanced Design with CSS

(12 Lectures)

Review of CSS. CSS Selectors. Specificity and the cascade. Pseudo-classes and pseudo-elements. CSS Box sizing. Gradient and Transparent backgrounds. CSS Typography – Line height and letter spacing. Web-safe fonts. CSS Layout. Display property – inline, block, inline-block, none. Positioning – static, relative, absolute, fixed. Floats and clear property. Box alignment – flexbox and grid layout. Responsive Web Design – Media queries and breakpoints. Fluid layouts. Brief concept of CSS preprocessors – Sass, Less. Brief concept of CSS frameworks – Bootstrap, Tailwind.

Unit 3: Advanced JavaScript

(12 Lectures)

Review of JavaScript concepts. Functions in JavaScript. Lexical Environment. Arrays and Array manipulation in JavaScript.. JavaScript Events and Event Handling – Event propagation and event delegation. Implementing interactivity with user actions. Introduction to JavaScript APIs. Callback functions and event loop. Promise chain. Asynchronous function with async/wait. DOM manipulation and event handling with jQuery. Overview of AJAX. Brief concept of XMLHttpRequest object.

Unit 4: Server-Side Scripting using PHP

(10 Lectures)

Review of PHP as a server-side scripting language. Handling forms and user input with PHP. Interacting with databases and performing CRUD operations using PHP and MySQL. User authentication using PHP. Implementing user registration and login functionality. Session management and Token based authentication. Overview of Cookies and their use in Web applications. Working with cookies in PHP – setting, reading, deleting. Concept of Cross-site scripting (XSS).

Unit 5: Advanced Concepts of Web Programming

(5 Lectures)

Overview of web hosting – shared hosting, VPS, dedicated hosting, cloud hosting. Overview of Server-Side Includes (SSI). Brief concepts of Web APIs and data integration. Concept of JavaScript frameworks – React.js and Node.js. Version Control Systems. Brief overview of Continuous Integration and Deployment. Overview of Web security and SSL/TLS. Web analytics and monitoring.

b) List of Practical

(This is a suggestive list only. Questions need not be restricted to this list.)

1. Create a semantic HTML structure for a blog post, including headings, paragraphs, images, and nested elements.
2. Develop an HTML5 video player with custom controls, including play, pause, volume control, and full-screen functionality.
3. Create a responsive HTML layout using CSS Grid or Flexbox that adapts to different screen sizes and orientations.
4. Develop a responsive navigation menu that collapses into a hamburger menu for mobile devices, utilizing media queries and CSS transitions.
5. Implement a CSS animation or transition to create a smooth fade-in effect for an element on page load.
6. Design a CSS grid layout that displays a multi-column card-based UI, where each card has a consistent height but variable width. Each card should display an image, title, and description.
7. Develop a CSS-only tooltip that appears when hovering over an element, with customizable styles and positioning.
8. Design a CSS drop-down menu with multiple levels of nested submenus, allowing users to navigate through the menu hierarchy.
9. Create a CSS layout that implements a sticky header, where the header remains fixed at the top of the page while the content scrolls.
10. Build a responsive landing page using HTML5, including a hero section, feature sections, and a contact form.
11. Implement a CSS grid-based layout for a product catalog, showcasing multiple products with consistent spacing and alignment.
12. Implement a custom dropdown menu using HTML, CSS, and JavaScript, with options that can be selected and displayed.

13. Build a form validation mechanism using HTML5 form validation attributes and JavaScript, ensuring that required fields are filled out correctly. Use CSS to design the form and the validation messages.
14. Develop a slideshow or carousel using JavaScript and the DOM API, with next/previous controls and automatic playback.
15. Implement a dynamic table that allows users to add or remove rows, with the ability to edit and delete individual cells.
16. Develop a live search functionality that filters and displays search results from the content of the web page in real-time as the user types, using JavaScript and DOM manipulation.
17. Use a callback function to perform an asynchronous AJAX request and update the content of a specific HTML element with the response.
18. Implement a callback-based timer that executes a specific function after a certain period of time has elapsed.
19. Create a simple asynchronous form submission process using AJAX, displaying a loading spinner while waiting for the response.
20. Develop a weather application that uses an asynchronous API call to fetch weather data based on user input, displaying the results on the page.
21. Implement a user registration form in PHP, which securely stores user credentials in a database and performs validation checks for email uniqueness and password strength.
22. Create a login page in PHP that verifies user credentials against the stored data in the database and redirects authenticated users to a secure dashboard.
23. Develop a Password reset functionality in PHP, allowing users to request a password reset link via email and securely update their password.
24. Implement a user profile page in PHP, which displays and allows users to edit their personal information such as name, email, and profile picture.
25. Create a session-based shopping cart system in PHP, allowing users to add products, update quantities, and remove items, while maintaining cart information across different pages.
26. Develop an access control system in PHP, where certain pages or features are restricted to logged-in users only and unauthorized users are redirected to a login page.
27. Implement user roles and permissions in PHP, allowing administrators to assign different levels of access to users based on their roles (e.g., admin, moderator, user).
28. Create a "Remember Me" functionality in PHP, using cookies to remember and automatically log in returning users for a certain period of time.
29. Develop a logout mechanism in PHP that destroys the user session and redirects users to a logout confirmation page or the login page.
30. Implement account activation via email in PHP, where new users receive an activation link to verify their email address and activate their account.

Particulars of course designer:

Name : Risheraj Baruah

Contact No. : +91 8486942427

Email id : rishirajbaruah@gauhati.ac.in

Paper Name: Artificial Intelligence

1. Learning Outcome:

After completing this course, students will know the fundamentals of artificial intelligence (AI), identify problems where artificial intelligence techniques are applicable and able to apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- d) Theory: 45 hrs
- e) Practical: 30 hrs
- f) Non Contact: 5 hrs

9. List of Books:

- g) *Rich & Knight, Artificial Intelligence* – Tata McGraw Hill, 2nd edition, 1991.
- h) *Russell & Norvig, Artificial Intelligence-A Modern Approach*, LPE, Pearson Prentice Hall, 2nd edition, 2005.
- i) *W.F. Clocksin and Mellish, Programming in PROLOG*, Narosa Publishing House, 3rd edition, 2001.
- j) *DAN.W. Patterson, Introduction to A.I and Expert Systems* – PHI, 2007.
- k) *Ivan Bratko, Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 3rd edition, 2000.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction

(4 Hours)

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

UNIT 2: Problem Solving and Searching Techniques

(16 Hours)

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

UNIT 3: Knowledge Representation

(14 Hours)

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG)

UNIT 4: Dealing with Uncertainty and Inconsistencies

(6 Hours)

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

UNIT 5: Understanding Natural Languages

(5 Hours)

Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

Practical:

- Write a prolog program to calculate the sum of two numbers.
- Write a prolog program to find the maximum of two numbers.
- Write a prolog program to calculate the factorial of a given number.
- Write a prolog program to calculate the nth Fibonacci number.
- Write a prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the nth element into every list at all levels.
- Write a Prolog program to remove the nth item from a list.
- Write a Prolog program, remove_nth (Before, After) that asserts the After list is the Before list with the removal of every nth item from every list at all levels.
- Write a Prolog program to implement append for two lists.
- Write a Prolog program to implement palindrome (List).
- Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
- Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
- Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
- Write a Prolog program to implement two predicates evenlength(List) and oddlength (List) so that they are true if their argument is a list of even or odd length respectively.
- Write a Prolog program to implement reverse (List, Reversed List) that reverses lists.
- Write a Prolog program to implement maxlist (List, Max) so that Max is the greatest number in the list of numbers List using cut predicate.
- Write a Prolog program to implement GCD of two numbers.
- Write a prolog program that implements Semantic Networks/Frame Structures.

Particulars of course designer:

Name: Dr. Diganta Kumar Pathak

Contact No.: 9707737222

E-mail id: digantakumarpathak@gauhati.ac.in

Paper Name: Computer Graphics

1. Learning Outcome:

After completing this course, students will know about basic elements of Computer Graphics, fundamental of Computer graphics algorithms along with basic mathematical foundations of computer graphics.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) D. Hearn, M. Baker: Computer Graphics, Prentice Hall of India 2008.
- b) J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice 2nd edition Publication Addison Wesley 1990.
- c) D.F.Rogers Procedural Elements for Computer Graphics, McGraw Hill 1997.
- d) D.F.Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill, 2nd edition 1989.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Introduction

(2 Hours)

Basic elements of Computer Graphics, Applications of Computer Graphics

UNIT 2: Graphics Hardware

(5 Hours)

Input Devices: Keyboard, Mouse, Trackball & Space ball, Joystick, Data Glove, Digitizers, Image Scanners, Touch panels, Light Pens systems. Output display devices: Refresh CRT, Raster-Scan display and Random-scan display technique, Color display techniques-Beam penetration method and Shadow-mask method, Direct view storage tubes, Emissive & Non-emissive flat-panel, Displays-Plasma panels, LED and LCD monitor, Three-dimensional viewing devices and Virtual-Reality systems Display processor: Raster-scan systems, Random-scan systems

UNIT 3: Fundamental Techniques in Graphics

(20 Hours)

Line-drawing algorithms:DDA algorithm and Bresenham's Line drawing Algorithm, Midpoint Algorithm for Circle and Ellipse Generation, Curve generation. Attributes for output primitives: Area-filling Algorithms - Scan-line Polygon-fill, 2-D Geometric Transformations: Basic transformations-translation, Rotation and Scaling Matrix representations and Homogeneous Co-ordinate representations, Composite transformations among translation, Rotation and Scaling, 2-D viewing: Definition, Viewing transformation pipeline, Window-to-viewport Co-ordinate transformation. 2-D Clipping: Concept and Algorithm: Point clipping, Line clipping - Cohen-Sutherland algorithm, Area clipping, Text clipping, Polygon clipping. 3-D concepts: Display methods-Parallel projection, perspective projection 3-D geometric transformations: Transformation, Translation, Rotation and Scaling around axes, 3-D Viewing Projections – Parallel and Perspective.

UNIT 4: Geometric Modelling**(8 Hours)**

Representing curves and surface, Bezier curves and surfaces – Definition of Bezier curve and its properties, Algorithms for Bezier curves and surfaces, Hermite curve

UNIT 5: Visible Surface determination**(5 Hours)**

Definition, approaches for visible surface detection, object-space methods- Back-Face Detection, Image space methods: Depth Buffer Methods, A Buffer Method, Scan Line Method, Depth-Sorting Method

UNIT 6: Surface rendering**(5 Hours)**

Definition and importance, light sources, Basic illumination models-Ambient light, Diffuse reflection, Specular reflector and Phong model

Practical:

- Write a program to implement DDA algorithm for line drawing.
- Write a program to implement Bresenham's line drawing algorithm.
- Write a program to implement mid-point circle drawing algorithm.
- Write a program to clip a line using Cohen-Sutherland line clipping algorithm.
- Write a program to clip a polygon using Sutherland Hodgeman algorithm.
- Write a program to apply 2D translation on a 2D object (use homogenous coordinates).
- Write a program to apply 2D rotation on a 2D object (use homogenous coordinates).
- Write a program to apply 2D scaling on a 2D object (use homogenous coordinates).
- Write a program to apply 2D reflection of a 2D object (use homogenous coordinates).
- Write a program to apply 2D shear operation on a 2D object (use homogenous coordinates).
- Write a program to apply 3D translation on a 3D object (use homogenous coordinates).
- Write a program to apply 3D rotation on a 3D object (use homogenous coordinates).
- Write a program to apply 3D scaling on a 3D object (use homogenous coordinates).
- Write a program to apply 3D reflection of a 3D object (use homogenous coordinates).
- Write a program to apply 3D shear operation on a 3D object (use homogenous coordinates).
- Write a program to draw Hermite/Bezier curve.

Particulars of course designer:

Name: Dr. Diganta Kumar Pathak

Contact No.: 9707737222

E-mail id: digantakumarpathak@gauhati.ac.in

Paper Name: Data Mining and Warehousing

1. Learning Outcome:

- a) Understanding the process of Knowledge Discovery in Databases.
- b) Understand the functionality of the various data warehousing component.
- c) Characterize the kinds of patterns that can be discovered by association rule mining.
- d) Analysis of different types of data by clustering and classification.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- a) Theory: 45 hrs
- b) Practical: 30 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- a) A.K. Puzari, *Data Mining Techniques*, University Press.
- b) J. Han, J. Pei and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann.
- c) P. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, Pearson Education (LPE).
- d) G. K. Gupta, *Introduction to Data Mining with Case Studies*, PHI.

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

UNIT 1: Overview

(4 Lectures)

What is Data Mining?, Knowledge Discovery in Databases (KDD) vs. Data Mining, Types of Data, Basic Data Mining Tasks, Predictive and Descriptive data mining techniques, Supervised and Unsupervised learning techniques, Basics of Pre-processing methods- Data Cleaning, Data Integration and Transformation, Data Reduction, Data Visualization.

UNIT 2: Data Warehousing

(6 Lectures)

What is Data Warehouse? Multidimensional Data Model, Data Cube, Basic Components of Multidimensional Data Model, OLAP Operations- Slicing, Dicing, Drilling, Drill-Up, Drill-Down, Drill-Within, Drill-Across, Pivot(Rotate), Schema of Warehouse, Data Warehouse Architecture, Metadata.

UNIT 3: Association Rule Mining

(12 Lectures)

What is Market Basket Data?, k-Itemset, Support of an Itemset, Frequent Itemsets, Infrequent Itemsets, Maximal Frequent Itemsets, Closed Frequent Itemsets, Association Rules, Confidence of a Rule, Problem of Mining Association Rules, Algorithm for Mining Frequent Itemsets- Apriori Algorithm, Pincer-Search Algorithm, DIC (Dynamic Itemset Counting) Algorithm, Steps of Mining Association Rules.

UNIT 4: Clustering

(12 Lectures)

What is Clustering, Partitional vs Hierarchical Clustering, Types of Data in Clustering, Distance Measures used in Clustering- Euclidean Distance, Manhattan Distance, Similarity Measures used in

Clustering- Cosine Similarity, Jacquard Coefficient, Partitional Clustering Methods- K-Means, K-Medoids, PAM, CLARA, CLARANS, Density Based Clustering Methods- DBSCAN, Introduction to Hierarchical Clustering.

UNIT 5: Classification

(8 Lectures)

What is Classification? Issues Regarding Classification, K-Nearest Neighbor Classifiers, Bayesian classification, Introduction to Decision Tree.

UNIT 6: Recent Trends and Techniques used in Data Mining

(3 Lectures)

Basic Concepts of- Web Mining, Spatial Data Mining, Temporal Data Mining, Big Data Mining, Concept of Neural Network, Genetic Algorithm.

Practical / Lab work to be performed

- Implement **any one** from the following-
 - Write a computer program to implement A priori algorithm to mine all frequent itemsets from a transactional dataset. Use hashing to store the item sets in the level wise generation of candidate sets.
 - Write a computer program to implement the Pincer Search algorithm.
 - Write a computer program to implement the DIC (Dynamic Item set) algorithm.
- Implement **any four** from the following-
 - Write computer program to implement the K-Means algorithm using different distance measures stated in the syllabus.
 - Write computer program to implement the PAM algorithm using different similarity measures stated in the syllabus.
 - Write a computer program to implement the CLARA algorithm.
 - Write a computer program to implement the CLARANS algorithm.
 - Write a computer program to implement the DBSCAN algorithm.
 - Write a computer program to implement the K-NN algorithm.

Particulars of Course Designer:

Name: Dr Irani Hazarika

Contact No: 8486965773

Email: queensarathi@gmail.com

Paper Name: Graph Theory

1. Learning Outcome:

- After completing this course, students will have understanding of graph theoretic concepts, problems and associated algorithmic solutions.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 4

7. Practical Credit: 0

8. No of Hours:

- a) Theory: 60 hrs
- b) Practical: 0 hrs
- c) Non Contact: 5 hrs

9. List of Books:

- e) *Introduction to Graph Theory*, Douglas B. West, Pearson
- f) *Introduction to Graph Theory*, Robin J. Wilson, Pearson Education Limited
- g) *Graph Theory with Applications to Engineering and Computer Science*, Narasingh Deo, PHI

10. Contents of Syllabus:

A. Theory

Detailed Syllabus:

Unit I: Introduction

5 hrs

Graph, directed and undirected graph, weighted and unweighted graph, simple and multigraph, degree, in degree and out degree, Handshaking theorem, complete graph, bipartite graph, cut set, cut vertices, graph representations: incidence matrix, adjacency matrix and adjacency list, BFS traversal and DFS traversals on a graph using stack and queue data structures, isomorphism, homomorphism

Unit II: Connectivity, paths and cycle

15 hrs

Walk, path and cycle, connected graphs, disconnected graphs, components, Hamiltonian path, Hamiltonian cycle, Hamiltonian graphs, Dirac's theorem, Eulerian path, Eulerian cycle, Euler graphs, Fleuri's algorithm, 2-connected graphs, connectivity and digraph, k-connected and k-edge connected graphs, application of Menger's theorem, Shortest path problem, variations of shortest path problem: single source shortest path problem, single pair shortest path problem and all pairs shortest path problem, Dijkstra's algorithm, Bellman Ford algorithm, Floyd Warshall's algorithm, Johnson's algorithm

Unit III: Tree

12 hrs

Tree, forest, properties of tree, spanning tree, spanning forest, counting trees, Cayley's theorem, matrix-tree theorem, minimum spanning tree, Kruskal's algorithm, Prim's algorithm, disjoint spanning trees, graph decomposition, graceful labeling, graceful graph, binary tree, binary search tree, AVL tree, multiway search tree, B tree, B+ tree

Unit IV: Matching and coloring

13 hrs

Matching, bipartite matching, maximum bipartite matching, Ford Fulkerson's algorithm for finding maximal bipartite matching, perfect bipartite matching, non-bipartite matching, maximal non-bipartite matching, largest maximal matching, perfect non-bipartite matching, Hall's Marriage theorem, vertex cover, vertex cover and matching, independent sets, dominating sets, atable

matching, Hungarian algorithm, introduction to Edmonds Blossom shrinking algorithm, vertex coloring, k-colorable graph, chromatic number, Brook's theorem, clique number, map coloring problem

Unit V: Digraph

7 hrs

Digraph, simple digraph, connected and strongly connected digraph, orientable graph, Eulerian digraph, Hamiltonian digraph, tournament, Markov chains, Flow networks, residual graph, augmenting path, Ford Fulkerson's algorithm

Unit VI: Classical problems

8 hrs

Travelling Salesman Problem, variants of Travelling Salesman Problem, Chinese Postman Problem, variants of Chinese Postman Problem, the minimum connector problem, Huffman coding and Huffman tree, Konigsberg bridge problem, three utilities problem

Particulars of course designer:

Name: Dr. Hasin Afzal Ahmed

Contact No.: 8011810533

E-mail id: hasin@gauhati.ac.in

Paper Name: Information Security and Cyber Laws

1. Learning Outcome:

After the completion of the course, the students will be able to develop basic understanding of security, cryptography, system attack and defences against them.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 4

7. Practical Credit: 0

8. No of Hours:

d) Theory: 60 hrs

e) Practical: 0 hrs

f) Non Contact: 5 hrs

9. List of Books:

- e) Merkow, M., & Breithaupt, J.(2005) Information Security Principles and Practices. 5th edition. Prentice Hall.
- f) Cryptography And Network Security Principles And Practice, Fourth or Fifth Edition, William Stallings, Pearson Edition.
- g) Cyber Law & Cyber Crimes, Advocat Prashant Mali; Snow White publications, Mumbai
- h) The Information Technology Act, 2000; Bare Act – Professional Book Publishers, New Delhi

10. Contents of Syllabus:

UNIT 1: Introduction

(15 Lectures)

Basic components of security (Confidentiality, Integrity and Availability), Attacks, Computer Crime, Security Services, Security Mechanism, Cyber Crimes, information Technology ACT, Cryptography, Substitution Cipher, Transposition Cipher, Block Cipher, Stream Cipher, Confusion, Diffusion, Symmetric Key, Asymmetric Key, Encryption, DES Algorithm, Hash Function, Digital Signature, Digital Certificate.

UNIT 2: Program Security

(10 Lectures)

Program Security, Program Errors, Buffer Overflow, Incomplete mediation, Time-of-check to Time-of-use Errors, Malicious codes, Virus, Threats, Control against Programs, Program Security Issues. Protection in OS: Memory and Address protection, Access control, File protection, User Authentication.

UNIT 3: Database Security

(10 Lectures)

Reliability, Integrity, Sensitive Data, Inference, Multilevel Security, Issues regarding the right to access information: Protecting Data, Multiple security level and categorization of data and users, Loss of integrity, Loss of availability, Loss of confidentiality, Access control, Inference control, flow control, data encryption

UNIT 4: Security in Networks (Cyber Attack)

(15 Lectures)

Threats in Networks, Security Controls- Architecture, Encryption, Content Integrity, Strong Authentication, Firewalls: Design and Types of Firewalls, Intrusion Detection System, Secure Email, Denial-of-service attacks, Man in the middle Attack, Phishing, Spoofing and Spam Attacks, Drive-by attack, SQL Injection, Birthday attack, Social Engineering attack, Password Attack. Cross site

scripting Attack, Malware Attack, Administering Security, Security Planning, Risk Analysis, Organisational Security Policy, Web Servers and Browsers, HTTP, Cookies, Caching, Secure Socket Layer (SSL), Secure Electronic Transaction (SET), E-mail Risks, Spam, E-mail Protocols, Simple Mail Transfer Protocol (SMTP), Post office Protocol (POP), Internet Access Message protocol (ICMP), Secured Mail: Pretty Good Privacy (PGP), S/MIME (Secure/Multipurpose Internet Mail Extensions)

UNIT 5: Cyber Laws

(10 Lectures)

Cyber crime, Types of crimes, Information technology Act 2000: Salient Feature of IT Act 2000, various authorities under IT Act and their powers, Penalties & Offences, amendments, Sections under the Information Technology Act such as:

- [Section 43] Penalty and compensation for damage to computer etc.
- [Section 65] Penalty for tampering with the computers source documents
- [Section 66] Punishment for hacking with computer system, data alteration etc
- [Section 66A] Punishment for sending offensive messages through any communication services
- [Section 66B] Receiving stolen computer's resources or communication devices dishonestly
- [Section 66C] Punishment for identity theft
- [Section 66D] Punishment for cheating by impersonation by using computer resource
- [Section 66E] Punishment for violation of privacy
- [Section 66F] Punishment for cyber terrorism
- [Section 67] Punishment for publishing or transmitting obscene material in electronic form
- [Section 67A] Punishment for publishing or transmitting of material containing sexually explicit act, etc. in electronic form
- [Section 67B] Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc. in electronic form
- [Section 72] Breach of confidentiality and privacy

Particulars of course designer:

Name : Dr. Pranamika Kakati

Contact No.: 9864201965

E-mail id: pranamikakakati20@gmail.com

Paper Name: Mobile Application Development

1. Learning Outcome: After completing this course, students will know:

- Fundamentals of Mobile Application Development.
- Difference between Native and Cross Platform Applications. Pros and Cons of Each Approach.
- To Design and Build a Complete Native Android Application with Both UI and Backend.
- To Design and Build a Complete Cross Platform Application with Both UI and Backend using Flutter.

2. Prerequisites:

- Solid Understanding of an Object Oriented Programming Language like Java.
- Fundamental Idea of UI Design.

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

- g) Theory: 45 hrs
- h) Practical: 30 hrs
- i) Non Contact: 5 hrs

9. List of Books:

- i) Android Programming: The Big Nerd Ranch Guide - Bill Phillips, Chris Stewart, Kristin Marsicano, Brian Gardner
- j) Professional Android - Reto Meier, Ian Lake
- k) Android Documentation - <https://developer.android.com/>

10. Contents of Syllabus:

a) Theory

Unit I: Introduction to Mobile Application Development

5 hrs

Fundamentals of Mobile Application – Understanding Mobile Application Development Basics, Major Mobile OSs and their market share, Understanding Cross Platform and Native Application Development, The advantages and disadvantages of each approach, Components of a Mobile Application, Basic Design Principles of Mobile UI including Wireframing, Typography and Content Flow.

Unit II: Getting Familiar with Android

5hrs

Introduction to Android Operating System. History and Versions of Android. Understanding the Basics of Android Operating System including OS architecture, Anatomy of an Android Application(apk), learning about various approaches of Android Application development like Native Application Development using Java/Kotlin or Cross Platform Application Development with Flutter/React-Native/Ionic etc. In-depth understanding of each approach and their pros and cons.

Unit III: Getting Started with Native Android Application Development

10 hrs

Setting up Android Studio and getting familiarized with the IDE, Setting up JDK and Android Emulator, Creating the First App – Hello World App, Understanding various essential folders and files associated with an Android App stored inside *manifests*, *java* and *res* directories. Basic understanding about *Gradle*. Running the App for the first time. Getting started with USB Debugging at a physical Android Device. Understanding debugging facilities available with Android Studio. Getting Started with XML for Android UI Design, Learning Various UI Components of Android. Working with various UI resources like Images, Colors, Fonts etc. Creating a UI oriented App from Scratch. Working with layout switching in Portrait and Landscape mode. Providing functionality to an Android Application using Java, Understanding Android *Activity* and its lifecycle, various events associated with Activity Lifecycle – *onCreate()*, *onStart()*, *onPause()*, *onResume()*, *onStop()*, *onRestart()*, *onDestroy()* etc. Broadcast Receivers, Intent and Filters. Advanced Layouts in Android including ListView, CardView, RecyclerView etc., Fragments, Material Design in Android – Principles and Implementation, Styles and Themes.

Unit IV: Advanced Android Application Development

10 hrs

Working with System Components in Android – File System Access, Location based Services, Phone, SMS, Bluetooth, Camera, Sensors etc. and App Permission Management, Working with Multimedia Content like Audio & Video in Android. Working with API Calls and Web Services, Packaging and Publishing Android Applications.

Unit V: Working with Databases in Android

5 hrs

Building database driven Apps in Android, Working with SQLite, Interacting with Remote Databases using JSON, Performing CRUD operations in both Local and Remote Databases. Understanding Realtime Databases and getting started with Firebase. Implementing Firebase backend in previously developed CRUD application.

Unit VI: Cross Platform Mobile Application Development using Flutter

10 hrs

Getting Started with Flutter and Dart, Understanding Flutter Architecture, Considering other alternatives, setting up the Development Environment, Material Design and System Services. Working with CRUD and HTTP Requests, Publishing and Packaging Apps for both Android and iOS and publishing at different platforms.

b) Practical Assignments

- 1) Build a Calculator App in Android.
- 2) Build a Tic-Tac-Toe Game. The game should keep records of Each Player and Game Time of each match.
- 3) Build an Android News Reader app which fetches news from an online API like Google News and shows the stories in a list. Whenever the user clicks on the heading of a particular story, the full story appears with the featured image.
- 4) Build a Simple Chat App in Flutter using Firebase. Export the app to both Android and iOS.

Particulars of course designer:

Name: Mr. Nibir Borpuzari

Contact No.: 8822306808

E-mail id: nibir@gauhati.ac.in

Paper Name: Optimization Techniques

1. Learning Outcome:

- On successful completion of the course, students will be able to get thorough knowledge on formulation of optimization model and solution methods on optimization.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 4

7. Practical Credit: 0

8. No of Hours:

- d) Theory: 60 hrs
- e) Practical: 0 hrs
- f) Non Contact: 5 hrs

9. List of Books:

- h) *Introduction to Graph Theory*, Douglas B. West, Pearson
- i) *Introduction to Graph Theory*, Robin J. Wilson, Pearson Education Limited
- j) *Graph Theory with Applications to Engineering and Computer Science*, Narasingh Deo, PHI

10. Contents of Syllabus:

Detailed Syllabus:

UNIT 1: Introduction

(5 Lectures)

Concept of Optimization- classification of optimization –problems, Simulation of Models, Art of Modeling.

UNIT 2: Modelling with Linear Programming

(10 Lectures)

Linear Programming Model, Two variable LP Model, Types of Formulation of Simplex Method, Dual Simplex Method, Sensitivity Analysis, LP Model in Equation Form, Transportation Problem, Network Model, Minimal Spanning Tree Algorithm, Shortest route Problem. Necessary and sufficient conditions for finding extrema point, Bisection method.

UNIT 3: Queuing Theory

(10 Lectures)

Queuing Model, Elements of Queuing Model, Pure Birth and Death Models, Queues with combined arrival and departures, random and series queues, Generalized and Specialized Queuing Models.

UNIT 4: Unconstrained Optimization

(10 Lectures)

Newton and Quasi-Newton methods, Conjugate gradient methods, Linesearch and Trust Region methods Quadratic programming problem-Wolfe’s method & Beale’s method.

UNIT 5: Constrained Optimization

(10 Lectures)

Linear programming, Equality and inequality linear constraints. Barrier and augmented Lagrangian methods, Sequential quadratic programming, Infeasible start Newton method, Interior-point methods (inequality constrained minimization; barrier method; primal-dual interior point method Goal

Programming-Basics of goal programming, goal programming formulation, goal programming algorithms: Weights method & preemptive method, Graphical solution

Particulars of course designer:

Name : Dr. Pranamika Kakati

Contact No.: 9864201965

E-mail id: pranamikakakati20@gmail.com

Paper Name: System Software

1. Learning Outcome:

After completing this course, students will have understanding of various types of system software.

2. Prerequisites: NIL

3. Semester: 6

4. Course Type: Elective

5. Course Level: 300-399

6. Theory Credit: 3

7. Practical Credit: 1

8. No of Hours:

j) Theory: 45 hrs

k) Practical: 30 hrs

l) Non Contact: 5 hrs

9. List of Books:

l) System Software : An Introduction to Systems Programming, Leland L. Beck, D. Manjula, Pearson

m) Systems Programming, Dhananjay Dhamdhare, McGraw Hill Education

10. Contents of Syllabus:

Unit I: Introduction to Operating System

10 hrs

Types of software, Application software and system software, examples of system software, system programming, system software and machine architecture, the simplified instructional computer (SIC): *memory, registers, data formats, instruction formats, addressing modes, instruction set, input and output*, programming examples in SIC\

Unit II: Assemblers

12 hours

Assembler definition, basic assembler functions, assembler algorithm and data structure, handling instruction formats and addressing modes, program relocation, handling literals, symbol defining statements, expressions, assembler design options: one pass assemblers and multi pass assemblers, introduction to NASM assembler

Unit III: Loaders and Linkers

7 hours

Loading, relocation and linking, loader, absolute loader, bootstrap loader, relocating loader, program linking, linking loader, linkage editor, static and dynamic linking

Unit IV: Macro processor

6 hours

Definition of macro processor, macro definition and expansion, macro processor algorithm and data structures, conditional macro expansion, general purpose macro processors, macro processing within language translators

Unit V: Compilers

10 hours

Compiler definition, grammars, lexical analysis, syntactic analysis, operator precedence parsing, recursive descent parsing, code generation, intermediate form, code optimization: machine dependent and machine independent, interpreter

(b) Practical

- 1) Introduction to NASM assembler (1 class/2 hrs)
- 2) Introduction to segments and registers (1 class/2 hrs)
- 3) A simple assembly program to print hello (1 class/2 hrs)
- 4) Input and output in assembly language (1 class/2 hrs)
- 5) Conditional statements in assembly language (2 classes/4 hrs)
- 6) Looping in assembly language (3 classes/6 hrs)
- 7) An assembly language program that accepts two numbers from the user and displays sum of the numbers (1 class/2 hrs)
- 8) An assembly language program that changes case of accepted characters (1 class/2 hrs)
- 9) An assembly program that accepts a number and displays whether the number is odd or even (1 class/2 hrs)
- 10) An assembly program that accepts a number n from the user and displays “hello world” n number of times. (1 class/2 hrs)
- 11) An assembly program that accepts a number from the user and displays factorial of the number (1 class/2 hrs)
- 12) An assembly program that accepts a number n from the user and displays whether the number is prime (1 class/2 hrs)

Particulars of course designer:

Name: Dr. Hasin Afzal Ahmed

Contact No.: 8011810533

E-mail id: hasin@gauhati.ac.in

1

INS-101 (INS-RC-1016)

Basic Circuit Theory and Network Analysis

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- None

Course Outcomes

- CO1: Describe the basic network theorem.
- CO2: Explain the basics of analog electronics (BJT, FET, MOSFET)
- CO3: Describe boolean algebra, combinational and sequential logic.

A Theory

A.1 Unit I : *Circuit Concepts and Circuit Analysis*

Voltage and Current Sources, Resistors.

Inductors: Fixed and Variable inductors, Self and mutual inductance.

Capacitors: Principles of capacitance, Parallel plate capacitor, Permittivity, Definition of Dielectric Constant, Dielectric strength, Energy stored in a capacitor, Air, Paper, Mica, Teflon, Ceramic, Plastic and Electrolytic capacitor, capacitors in series and parallel.

Kirchhof's Current Law (KCL), Kirchhof's Voltage Law (KVL), Node Analysis, Mesh Analysis.

RC Circuit, RL Circuit, RLC Circuits.

Sinusoidal Voltage and Current, AC/DC power source and power distribution. Definition of Instantaneous, Peak, Peak to Peak, Root Mean Square and Average Values. Voltage-Current relationship in Resistor, Inductor and Capacitor.

Passive Filters: Low Pass, High Pass, Band Pass and Band Stop.

A.2 Unit II : *Network Theorems*

Principal of Duality, Superposition Theorem, Thevenin's Theorem, Norton's Theorem, Reciprocity Theorem, Millman's Theorem, Maximum Power Transfer Theorem.

A.3 Unit III : Analog Electronics

PN Junction diode and device power rating, Basic transistor action, Transistor current components and amplification. Transistor configurations: Common Base (CB), Common Emitter (CE) and Common Collector (CC) configuration, I-V characteristics.

Concept of feedback, negative and positive feedback, Negative feedback, advantages and disadvantages of negative feedback, Barkhausen criteria for oscillations.

Junction Field Effect Transistor (JFET), Construction of JFET, Construction of MOSFET.

A.4 Unit IV : Digital Electronics

Decimal, Binary, Hexadecimal and Octal number systems, base conversions, Truth Tables of OR, AND, NOT, XOR, XNOR, Universal (NOR and NAND) Gates, Basic postulates and fundamental theorems of Boolean algebra, Combinational Logic Analysis and Design, Adder, Subtractor, Encoder and Decoder, Multiplexers and Demultiplexers, Sequential logic design, Latches and Flip flops , S-R Flip flop, J-K Flip flop, T and D type Flip flops, Introduction to registers and counters.

B Basic Circuit Theory and Network Analysis Lab

1. Familiarisation with
 - (a) Resistance in series, parallel and series-parallel. Type, wattage, tolerance, and temperature coefficient.
 - (b) Capacitors- Tolerance, voltage rating, type of capacitors, capacitors & inductors in series & parallel.
 - (c) Multimeter (Analog and Digital) - Checking of components.
 - (d) Voltage sources in series, parallel and series-parallel.
 - (e) Voltage and current dividers.
2. To study the Half wave rectifier and Full wave rectifier.
3. To study power supply using Zener diode and regulated power supply.
4. To verify and design AND, OR, NOT and XOR gates using NAND gates.
5. Design a Half and Full Adder.
6. Design a Half and Full Subtractor.
7. Flip Flop Type and its uses.

C Suggested books

1. Robert L. Boylestad, Louis Nashelsky, Electronic Devices and Circuit Theory, Prentice-Hall.
2. Albert Paul Malvino, Donald P. Leach, Digital principles and applications, McGraw-Hill.
3. Horowitz and Hill: Art of electronics, Cambridge University Press.
4. R. F. Coughlin and Driscoll, Op-amps and linear ICs, Prentice Hall.
5. A.P.Godse, U.A.Bakshi, Electronics Devices and Circuits, Technical Publications.
6. Millman and Halkias, Electronic devices and circuits, McGraw-Hill.

2

INS-151 (INS-HC-2026) Transducers and Sensors

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Class XII Mathematics and Physics
- Idea of units and measurement

Course Outcomes

- CO1: Describe the generalised instrumentation system and its different characteristics.
- CO2: Explain different types of errors associated with measurement systems.
- CO3: Categorise and compare different sensors and transducers and illustrate its applications.

A Theory

A.1 Unit I : Instrumentation & Measurement Systems

Generalised instrumentation systems block diagram representation, scope of instrumentation in Industrial organisation.

Measurement systems: static (accuracy, sensitivity, linearity, precision, resolution, threshold, range, hysteresis, dead band, backlash, drift), impedance matching and loading, dynamic characteristics (types, fidelity, speed of response, dynamic error).

A.2 Unit II : Errors

Systematic errors, instrumental errors, environmental errors, random errors, loading errors, source of errors in measuring instruments, Uncertainties types, propagation of uncertainties.

A.3 Unit III : Transducers

Classification- Active, Passive, Mechanical, Electrical, their comparison. Selection of Transducers, Principle and working of following types: Displacement transducers - Resistive (Potentiometric, Strain Gauges - Types, Gauge Factor, bridge circuits, Semi-conductor strain gauge) Capacitive (diaphragm), Inductive (LVDT-Principle and characteristics, Hall effect sensors).

A.4 Unit IV : Sensors

Piezoelectric (Element and their properties, Piezo Electric coefficients Equivalent circuit), light (photo -conductive, photo emissive, photo voltaic, semiconductor, LDR), Temperature (electrical and non-electrical) Pressure (load cell).

B Transducers and Sensors Lab

1. Measurement of pressure, strain and torque using strain gauge.
2. Measurement of speed using Electromagnetic transducer.
3. Measurement of speed using photoelectric transducers
4. Measurement of angular displacement using Potentiometer.
5. Measurement of displacement using LVDT.
6. Measurement using load cells.
7. Measurement using capacitive transducer.
8. Measurement using inductive transducer.
9. Measurement of Temperature using Temperature Sensors/RTD.
10. Characteristics of Hall effect sensor.
11. Measuring change in resistance using LDR.

C Suggested books

1. Doebelin & Manek, Measurement Systems, McGraw Hill.
2. Nakra & Choudhary, Instrumentation Measurements and Analysis, Tata McGraw-Hill.
3. A.K. Sawhney, Electrical & Electronic Measurements & Instrumentation.
4. Rangan, Sarma, and Mani, Instrumentation- Devices and Systems, Tata-McGraw Hill.
5. H.S Kalsi, Electronic Instrumentation, McGraw Hill.
6. DVS Murthy, Measurement & Instrumentation, PHI.
7. D. Patranabis, Sensors and Transducers, PHI.
8. Arun K. Ghosh, Introduction to Measurements and Instrumentation, PHI.

3

INS-201 (INS-HC-2016) Analog Devices and Circuits

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Class XII Mathematics

Course Outcomes

- CO1: Explain the properties of intrinsic and extrinsic semiconductors and analyse the behaviour of p-n junction diodes in clipper circuits, clamping circuits, and rectifiers.
- CO2: Analyse the basic operation of BJTs and identify the different configurations, including Common Base, Common Emitter, and Common Collector.
- CO3: Differentiate between negative and positive feedback and analyse the performance of voltage and current feedback amplifiers.
- CO4: Analyse the characteristics of JFETs, including the construction, working, and I-V characteristics, and explain the operation of MOSFET amplifiers.

A Theory

A.1 Unit I : *Semiconductor Basics*

Introduction to semiconductor materials, intrinsic & extrinsic semiconductors. p-n junction diode: Ideal diode, Formation of depletion layer, space charge at a junction, Diode Circuits: clipper circuits, clamping circuits. Half wave rectifier, Center tapped and bridge full wave rectifiers, calculation of efficiency and ripple factor. DC power supply: Block diagram of a power supply, Zener diode as voltage regulator, temperature coefficient of Zener diode.

A.2 Unit II : *The BJT*

Basic transistor action, Transistor current components and amplification. Transistor configurations: Common Base (CB), Common Emitter (CE) and Common Collector (CC) configuration, I-V characteristics and hybrid parameters, regions of operation, dc load line, Q point. CE amplifier: Self bias arrangement of CE, dc and ac load line analysis, Hybrid equivalent of CE, frequency response of CE amplifier.

A.3 Unit III : *Feedback Amplifiers*

Concept of feedback, negative and positive feedback, Negative feedback: advantages and disadvantages of negative feedback, voltage (series and shunt), current (series and shunt) feedback amplifiers, derivation of gain, input and output impedances for feedback amplifiers. Positive feedback: Barkhausen criteria for oscillations, Study of phase shift oscillator, Colpitts oscillator and Crystal oscillator.

A.4 Unit IV : *Junction Field Effect Transistor(JFET)*

Construction of JFET, idea of channel formation, pinch-off and saturation voltage, current-voltage output characteristics. Metal Oxide Field Effect Transistor (MOSFET): Basic construction and working.

B Analog Devices and Circuits Lab

1. To study the Half wave rectifier and Full wave rectifier.
2. To study power supply using C filter and zener diode.
3. To study Fixed Bias and Voltage divide Feedback configuration for transistor.
4. To design a Single Stage CE amplifier.
5. To study Class A, B and C Power Amplifier.
6. To study clipping circuits
7. To study clamping circuits
8. To study the Colpitt's Oscillator.
9. To study the Phase Shift Oscillator.
10. To study the frequency response of Common Source FET amplifier.

C Suggested books

1. R. L. Boylestad, L. Nashelsky, K. L. Kishore, *Electronic Devices and Circuit Theory*, Pearson Education.
2. Albert Paul Malvino, *Electronic Principles*, McGraw-Hill.
3. J. Millman and C. Halkias, *Integrated Electronics*, Tata McGraw Hill.
4. J. Cathey, *2000 Solved Problems in Electronics*, Schaum's outline Series, Tata McGraw Hill.

4

INS-251 (INS-HC-6026) Control Systems

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Applied Mathematics (of 1st semester) or equivalent

Course Outcomes

- CO1: Understand open loop and closed loop control systems, derive transfer function and apply basic control actions.
- CO2: Analyse transient and steady-state response of systems, design controllers with stability analysis and understand Root Locus plots.
- CO3: Correlate time and frequency response, design and analyse systems using Bode plots, Nyquist stability criterion, and gain and phase margins.

A Theory

A.1 Unit I : *Introduction to Control System*

Introduction of open loop and closed loop control systems, mathematical modelling of physical systems (Electrical, Mechanical and Thermal), derivation of transfer function, block diagram representation & signal flow graph, Reduction Technique, Mason's Gain Formula. Effect of feedback on control systems, Basic Control Actions: ON-OFF and Proportional.

A.2 Unit II : *Time Domain Analysis*

Time - Domain Analysis:-Time domain performance criteria, transient response of first, second & higher order systems, steady state errors and static error constants, performance indices, response with Proportional Controllers. Concept of Stability: Asymptotic stability and conditional stability, Routh - Hurwitz criterion, relative stability analysis, Root Locus plots and their applications.

A.3 Unit III : *Frequency Domain Analysis*

Frequency Domain Analysis: Correlation between time and frequency response, Polar and inverse polar plots, frequency domain specifications, Logarithmic plots (Bode Plots), gain and phase margins, Nyquist stability criterion, relative stability using Nyquist criterion, constant M & N circles.

B Control Systems Lab

1. To study characteristics of :
 - (a) Synchro transmitter receiver
 - (b) Synchro as an error detector
2. To study position control of DC motor
3. To study speed control of DC motor
4. To study time response of type 0,1 and 2 systems
5. To study frequency response of first and second order systems
6. To study time response characteristics of a second order system.
7. To study effect of damping factor on performance of second order system
8. Study of ON-OFF and Proportional Controllers.

C Suggested books

1. K. Ogata, *Modern Control Engineering*, Prentice Hall of India.
2. B. C. Kuo , *Automatic control system*, Prentice Hall of India.
3. J. Nagrath & M. Gopal, *Control System Engineering*, New Age International.

5

INS-252 (INS-HC-4016) Operational Amplifiers and Applications

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Class XII Physics
- Knowledge of Semiconductor devices
- Knowledge of BJT and UJT and their related parameters

Course Outcomes

- CO1: Students will be able to define opamps and its different parameters and configurations.
- CO2: Students will be able to explain the working of opamp based circuits.
- CO3: Students will be able to implement and make simple working circuits.

A Theory

A.1 Unit I : Operational Amplifier

Basic Operational Amplifier: Concept of differential amplifiers (Dual input balanced and unbalanced output), constant current bias, current mirror, block diagram of an operational amplifier (IC 741). Op-Amp parameters: input offset voltage, input offset current, input bias current, differential input resistance, input capacitance, offset voltage adjustment range, input voltage range, common mode rejection ratio, slew rate, supply voltage rejection ratio.

A.2 Unit II : Op-Amp Circuits

Op-Amp Circuits: Open and closed loop configuration, Frequency response of an op-amp in open loop and closed loop configurations, Inverting, Non-inverting, Summing and difference amplifier, Differential amplifier, Instrumentation Amplifier, Integrator, Differentiator, Voltage to current converter, Current to voltage converter. Comparators, Voltage limiters, Schmitt Trigger. Signal generators: Phase shift oscillator, Wein bridge oscillator, Square wave generator, triangle wave generator and Voltage controlled oscillator.

A.3 Unit III : Opamp Based Devices

Multivibrators (IC 555): Block diagram, Astable and monostable multivibrator circuit, Applications of Monostable and Astable multivibrators. Introduction to Phase locked loops (PLL): Block diagram, phase detectors, IC565.

Fixed and variable IC regulators: IC 78xx and IC 79xx -concepts only, IC LM317- output voltage equation.

A.4 Unit IV : Signal Conditioning circuits

Sample and hold systems, Active filters: First order low pass and high pass butterworth filter, Second order filters, Band pass filter, Band reject filter, All pass filter, Log and antilog amplifiers.

B Operational Amplifiers and Applications Lab

1. Study of op-amp characteristics: CMRR and Slew rate.
2. Designing of an amplifier of given gain for an inverting and non-inverting configuration using an op-amp.
3. Designing of analog adder and subtractor circuit.
4. Designing of an integrator using op-amp for a given specification and study its frequency response.
5. Designing of a differentiator using op-amp for a given specification and study its frequency response.
6. Designing of a First Order Low-pass filter using op-amp.
7. Designing of a First Order High-pass filter using op-amp.
8. Designing of a RC Phase Shift Oscillator using op-amp.
9. Study of IC 555 as an astable multivibrator.
10. Study of IC 555 as monostable multivibrator.
11. Designing of Fixed voltage power supply using IC regulators using 78 series and 79 series.

C Suggested books

1. R. A. Gayakwad, Op-Amps and Linear IC's, Pearson Education.
2. R. F. Coughlin and F. F. Driscoll, Operational amplifiers and Linear Integrated circuits, Pearson Education.
3. J. Millman and C.C. Halkias, Integrated Electronics, Tata McGraw-Hill.
4. A.P. Malvino, Electronic Principles, Tata McGraw-Hill.
5. K.L. Kishore, OP-AMP and Linear Integrated Circuits, Pearson.

6

INS-253 (INS-HE-5016) Signals and Systems

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Class XII Mathematics
- Knowledge of Differential and Integral Equations
- Knowledge of Fourier Transforms

Course Outcomes

- CO1: Students will be able to define discrete and continuous signals.
- CO2: Students will be able to explain the basics of time variant and invariant systems.
- CO3: Students will be able to solve and find the output of mentioned systems for given input.

A Theory

A.1 Unit I : *Signals and Systems*

Continuous and discrete time signals, Transformation of the independent variable, Exponential and sinusoidal signals, Impulse and Unit step functions, Continuous-Time and Discrete-Time Systems, Basic System Properties.

A.2 Unit II : *Linear Time-Invariant Systems (LTI)*

Discrete time LTI systems, the Convolution Sum, Continuous time LTI systems, the Convolution integral, Properties of LTI systems, Commutative, Distributive and Associative.

A.3 Unit III : *Properties of LTI System*

LTI systems with and without memory, Invariability, Causality (Causal systems), Stability, Unit Step response, Differential and Difference equation formulation, Block diagram representation of first order systems.

A.4 Unit IV : Laplace Transform

Laplace Transform, Inverse Laplace Transform, Properties of the Laplace Transform, Laplace Transform Pairs, Laplace Transform Methods in Circuit Analysis, Impulse and Step response of R, L and C circuits. Introduction to FIR systems.

B Suggested books

1. H. P. Hsu, Signals and Systems, Tata McGraw Hill.
2. S. T. Karris, Signal and Systems: with MATLAB, Computing and Simulink Modelling, Publications.
3. W. Y. Young, Signals and Systems with MATLAB, Springer.
4. M. Roberts, Fundamentals of Signals and Systems, Tata McGraw Hill.

C Signals and Systems Lab

1. Learning Scilab/MATLAB/Python (Experiments based on available system)
2. Explorations of Signals and Systems using Scilab/MATLAB/Python
3. Generation of Signals: continuous time
4. Generation of Signals: discrete time
5. Convolution of Signals
6. Solution of Difference equations.
7. Introduction to SIMULINK or open-source alternatives, and calculation of output of systems represented by block diagrams.

7

INS-254 (INS-HC-5026) Microprocessors

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Good knowledge on digital electronics
- Basic knowledge on programming techniques

Course Outcomes

- CO1: Describe internal mechanism of 8085 microprocessor system.
- CO2: Discuss the programming techniques of 8085 microprocessor.
- CO3: Illustrate different peripheral devices and interfacing techniques.
- CO4: Explain briefly the 8086 microprocessor and its interfacing.

A Theory

A.1 Unit I : 8085 Microprocessor

Introduction to 8085 Microprocessor, Pin description of 8085, Architecture, register of 8085, addressing mode. Instruction Type and Instruction Set, Machine Cycle, Instruction Cycle, Timing Diagram, Memory System, internal and external memory and concept of Virtual Memory. Hardware Interfacing or Types of I/O addressing-Interfacing Memory and Peripheral (I/o Mapped I/O and memory mapped I/O).

A.2 Unit II : Programming

Assembly Language Programming, Stacks and Subroutine, Interrupts of 8085-Hardware and Software interrupts. Difference between RISC and CISC Processor.

A.3 Unit III : Peripherals

Interfacing ICs- Programmable Peripheral Interface: Intel 8155, 8253, 8255, programmable Interrupt Controller: Intel 8259.

A.4 Unit IV : *Interfacing*

Application of Microprocessor 8085 in Instrumentation: Interfacing of Stepper Motor. Introduction to 8086 Microprocessor: keyboard Basics of 8086 (16 bit Microprocessor), Architecture of 8086, Concept of parallel processing in 8086.

B Microprocessors Lab

1. To write an assembly language program to perform basic mathematical operations (addition, subtraction, multiplication, division)
2. To write an assembly language program to generate first N terms of an A.P. / G.P. series
3. To write an assembly language program to generate first N terms of Fibonacci series
4. To write an assembly language program to arrange the given list of number in ascending / descending order
5. To write an assembly language program to calculate N!
6. To write an assembly language program to separate prime numbers in a given list of number
7. To write an assembly language program to convert a number from one number system to another

C Suggested books

1. Ramesh Gaonkar, Microprocessors architecture, programming and Applications, Wiley Eastern Ltd.
2. Liu Gibson, Microprocessor Systems: The 8086/8088 family Architecture, Programming & Design, PHI.
3. Barry B. Brey, The Intel Microprocessors, Pearson Education India.
4. Walter Triebel & Avtar A.Singh, 8088 and 8086 Microprocessors: Programming, Interfacing, Software Hardware and Applications, Pearson Education.
5. D. V. Hall, Microprocessors and Interfacing, Tata McGraw Hill.

9

INS-301 (INS-HC-4026) Analytical Instrumentation

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- A paper on Sensors and Transducers. In addition, a course on materials science or solid state physics or condensed matter physics will be advantageous.

Course Outcomes

- CO1: Students will gain an understanding of the theory and principles behind various spectro-analytical methods.
- CO2: Students will be able to discuss and explain the theory and principles behind various chromatographic separation methods.
- CO3: Students will be able to describe the fundamental principles and theory behind various diffraction techniques used in materials analysis.
- CO4: Students will be able to understand and identify the thermo analytical methods to be used for thermal analysis of materials.
- CO5: Students will learn about the instrumentation and applications of these methods in qualitative and quantitative analysis.

A Theory

A.1 Unit I : Analytical Instruments and Diffraction Techniques

Classification of instrumental methods, physical principles behind diffraction techniques, X-ray source and detectors, X-ray diffraction methods, single crystal diffraction, powder X-ray diffraction, elements of neutron diffraction, electron diffraction, applications of diffraction techniques.

A.2 Unit II : Atomic and Molecular Spectroscopy

Interaction of electromagnetic waves and matter; Colorimetry and Spectrophotometry ; Beer-Lambert's law; principles of atomic emission and absorption spectroscopy; rotational, vibrational and electronic transitions in molecules; optical systems used in spectroscopy - radiation sources, wavelength selectors, detectors; spectroscopic techniques and instruments used; comparison of atomic and molecular spectroscopy; applications.

A.3 Unit III : *Chromatography and Mass Spectrometry*

Principles of chromatography; instrumentation and applications of Thin layer chromatography (TLC). Column chromatography: Principle, process of elution through a column, chromatogram, band broadening, capacity factor, selectivity factor, Column efficiency, number of plates, plate height, column resolution. Gas Chromatography, instrumentation and applications; Principles of High Performance Liquid Chromatography (HPLC). Theory of mass spectrometer, components of mass spectrometer and applications.

A.4 Unit IV : *Thermal Analysis*

Introduction to thermal methods of analysis, thermo gravimetric analysis (TGA), differential thermal analysis (DTA), differential scanning calorimetry (DSC). Instrumentation and applications.

B Analytical Instrumentation Lab

1. Determination of pKa value for a dye using double beam spectrophotometer.
2. Spectrometric determination of iron in water sample using double beam spectrophotometer.
3. Determination of concentrations of sodium, calcium, lithium and potassium in sample using flame photometer.
4. Determination of concentration of potassium ions in sample by standard addition method using flame photometer
5. Spectrum interpretation using FT-IR.
6. Analysis of various ions using atomic absorption system.
7. Thin layer chromatographic (TLC) separation of samples from different origins (Biological / Pharmaceutical / Food).
8. Qualitative analysis of samples using Gas Chromatography.
9. Qualitative analysis of samples using High Performance Liquid Chromatography.

C Suggested books

1. Skoog & Lerry, Instrumental Methods of Analysis, Saunders College Publications.
2. H.H. Willard, Instrumental Methods of Analysis, CBS Publishers.
3. D.C. Harris, Quantitate Chemical Analysis, W.H. Freeman.
4. G. D. Christian, Analytical Chemistry, John Wiley & Sons.
5. Skoog, West and Holler, Analytical Chemistry, Saunders College Publications.
6. Vogel's Textbook of Qualitative Chemical Analysis, ELBS.
7. J.A. Dean, Analytical Chemistry Notebook, McGraw Hill.
8. John H. Kennedy, Analytical Chemistry: Principles, Saunders College Publication.
9. W. Kemp, Organic Spectroscopy, ELBS.
10. Frank Settle (ed.), Hand book of Instrumental Techniques for Analytical Chemistry, Prentice Hall.
11. James W. Robinson, Undergraduate Instrumental Analysis, CRC Press.

10

INS-302 (INS-HE-5036)

Communication systems

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Knowledge of different kind of signal waveforms and their properties
- Knowledge of Semiconductor devices
- Knowledge of Mathematical transformations

Course Outcomes

- CO1: Students will be able to define transmitter, receiver, channel etc. of communication system.
- CO2: Students will be able to define and explain various modes of communication.
- CO3: Students will be able to understand and design simple communication circuits.
- CO4: Students will be able to understand the basic principles and differences of different modulation schemes.

A Theory

A.1 Unit I : *Basic communication system*

Block diagram, Information source and input transducer, Transmitter medium (introduction to guided and unguided medium), Noise, Receiver, Destination, Necessity for modulation, Pass-band Signals, Base-band signals, Types of communication systems.

A.2 Unit II : *Analog Modulation*

Definition - AM waveforms - Frequency spectrum and band width - Modulation index - DSB-SC, SSB, Independent SB, Vestigial SB - Comparison and application of various AM schemes, Definition-Relationship between FM PM-Frequency deviation - Spectrum and transmission BW of FM, comparison of AM and FM systems.

A.3 Unit III : *Radio Transmitter and Receiver*

AM transmitters-High level and low level transmitters - SSB transmitters. FM transmitters - Block diagram. AM receivers - operation, SSB receiver-Block diagram, FM receivers - Block diagram, Discriminator circuit, Automatic gain control, TV transmission and reception (monochromatic and coloured)

A.4 Unit IV : Digital Communication

Pulse Analog Modulation: Sampling theorem, Errors in Sampling. Pulse Amplitude Modulation (PAM), Pulse Width Modulation (PWM) and Pulse Position Modulation (PPM). Generation and detection of PAM, PWM, PPM. Need for digital transmission, Quantizing, Uniform and Non-uniform. Quantization, Quantization Noise, Companding, Concept of Line encoding. Decoding, Regeneration, Transmission noise and Bit Error Rate, PCM. Concept of Time Division Multiplexing (TDM), Frequency Division Multiplexing (FDM) and CDMA

B Suggested books

1. G. Kennedy and B. Davis, Electronic Communication Systems, Tata McGraw Hill.
2. R. P. Singh and S. D. Sapre, Communication Systems: Analog and Digital, Tata McGraw Hill.
3. L. E. Frenzel, Communication Electronics: Principles and Applications, Tata McGraw Hill.
4. L. W. Couch II, Digital and Analog Communication Systems, Pearson Education.
5. T. G. Thomas and S. Chandra Sekhar, Communication Theory, Tata McGraw Hill.
6. L. Temes and M. E. Schultz, Schaum's outline of theory and problems of Electronic Communication.
7. H. Taub and D. Schilling, Principles of Communication Systems, Tata McGraw Hill.
8. W. Tomasi, Electronic Communication Systems: Fundamentals through Advanced, Pearson Education.
9. H. P. Hsu, Analog and Digital Communications, Tata McGraw Hill.
10. S. Haykin, Communication Systems, Wiley India.

C Communication systems Lab

1. Study of Amplitude Modulation and Demodulation.
2. Study of Frequency Modulation and Demodulation.
3. Study of Pulse Amplitude Modulation.
4. Study of Pulse Width Modulation.
5. Study of Pulse Position Modulation.
6. Study of Pulse Code Modulation.

11

INS-303 (INS-HE-6026)

Embedded System and Robotics

Total Lectures : 40 Credits : 4 (Theory : 03, Lab : 01)

Prerequisites:

- Programming in C
- Digital Electronics
- Microprocessor

Course Outcomes

- CO1: Program and debug basic microcontroller systems using 8051 architecture, including interfacing with external devices and memory, and using 8051 C programming language.
- CO2: Design and implement embedded systems using 8051 microcontroller, including interfacing with various input/output devices, memory and communication links such as UART, SPI and RS232.
- CO3: Understand the basics of robotics and how to integrate embedded systems in robotic applications.

A Theory

(Lectures 40)

A.1 Unit I : 8051 Microcontroller

Introduction to RISC and CISC microcontrollers, architecture of 8051 microcontroller, microcontroller hardware, program and data memory. External memory, counters, timers, serial data I/O, interrupts. Addressing modes, Instructions - data transfer instructions, logical, arithmetic, jump and call instructions. Programming using 8051 C.

A.2 Unit II : Embedded System Design

8051 interfacing with Keyboard, display Units (LED, 7-segment display, LCD), ADC, DAC, Stepper motor, Introduction to AVR family (Atmega328 and Arduino) and its architecture.

Interfacing and Communication Links Serial Interfacing: UART, SPI and RS232.

A.3 Unit III : Robotics

Definition of Robotics and its applications, overview of Robotics systems and components, role of Embedded Systems in Robotics, overview of Robot programming languages (C, Python and ROS-Robot Operating System).

Introduction to Robot kinematics and dynamics, forward and inverse kinematics, trajectory planning and control, sensors and actuators used in Robotics.

B Embedded System and Robotics Lab

1. Write a program to multiply two 16 bit unsigned numbers.
2. Write a program to add N 8 bit unsigned integer numbers.
3. Write a program to arrange the unsigned integer numbers in ascending/descending order.
4. Write a program for LED blinking in a predetermined fashion using 8051 microcontrollers.
5. Write a Program to OUT an 8 - bit value on a 8051.
6. Write a program for interfacing LCD display using 8051.
7. Write a program to convert an analog voltage to digital bits using 8051.
8. Write a program to convert a digital signal to analog signal using 8051.
9. Write a program for temperature sensor interfacing through serial port on 8051 and AVR (Arduino) microcontroller kits.
10. Write a program for PWM control of DC motor using 8051 and AVR (Arduino) microcontrollers.
11. Write a program to drive a stepper motor using 8051 and AVR (Arduino) microcontrollers.

C Suggested books

1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, *The 8051 Microcontroller and Embedded Systems*, Pearson Education.
2. Muhammad Ali Mazidi, Sepehr Naimi, Sarmad Naimi, *The AVR Microcontroller and Embedded Systems Using Assembly and C*,
3. Robotic Engineering - An Integrated Approach by Richard D Klafter, Thomas A. Chmielewski and Michael Negin (PHI).

12

INS-304 (INS-HG-4026)

Nuclear Instrumentation

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Either of Electronic Instrumentation, Analog Devices and Circuits or Operational Amplifiers and Applications.

Course Outcomes

- CO1: Students will be able to explain the principles and applications of gas-filled detectors and scintillation counters.
- CO2: Students will learn about the safe and effective use of nuclear techniques through reliable instrumentation for measurement, diagnostic and control.
- CO3: Students will be able to describe the principles behind solid state detectors.

A Theory

A.1 Unit I : Introduction

Nuclei and Nuclear Decay, Alpha particle, beta particle, gamma ray, neutron. Electron Capture and Internal Conversion, radioactive decay law, the nuclear level diagram, interactions of radiation with matter, Neutrino interactions, natural and man-made sources of radiation, units of radiation and radiation protection.

A.2 Unit II : Gas and Semiconductor Detectors

Gas filled detectors, ionization chambers, counters with gas amplification, proportional counters, and microstrip gas counters (MSC), GEM and MICROMEGAS counters, Resistive plate chambers. Semiconductor detectors- Si and Ge semiconductor detectors, other semiconductor detector materials.

A.3 Unit III : Scintillation Detectors

Scintillation detectors, Organic scintillators, Inorganic Scintillators, Photomultiplier Tubes and Photo detectors, Radiation spectroscopy with scintillators, Applications of Scintillators in nuclear physics, high energy physics and medicine, slow and fast neutron detection

A.4 Unit IV : Nuclear Medicine System

Nuclear electronics, pulse processing, pulse shaping counting and timing, single channel and multichannel pulse analysis. Electrostatic accelerators, Cyclotrons, Synchrotrons and Colliders, Linear accelerators, Secondary beams, applications of accelerators. Nuclear Reactors.

B Nuclear & Biomedical Instrumentation Lab

- (a) Characterization of bio potential amplifier for ECG signals.
- (b) Study on ECG simulator
- (c) Recording of EEG
- (d) Measurement of heart sound using electronic stethoscope.
- (e) Study of pulse rate monitor with alarm system
- (f) Determination of pulmonary function.
- (g) Study on ultrasound transducers based on medical system

C Suggested books

- (a) G. F. Knoll, Radiation Detection and Measurements, Wiley.
- (b) S. Tavernier, Experimental Techniques in Nuclear and Particle Physics, Springer

12

INS-351

Process Control Instrumentation

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Good knowledge on Control system engineering
- Good knowledge on Measurement Techniques

Course Outcomes

- CO1: Describe Process control principles and its different characteristics.
- CO2: Discuss controller principles.
- CO3: Explain different control modes.
- CO4: Illustrate PLC basics.

A Theory

A.1 Unit I : Introduction to Process control

Process control principles, servomechanisms, process control block diagram, Control system evaluation, stability, steady state response, Transient response, supervisory control, direct digital control, Final control operation- Pneumatic control, principles, amplification and conversion, pneumatic actuator.

A.2 Unit II : Controller Principles

Process Characteristics- process equation, process load, process lag, self-regulation, Control system parameters- Error, variable range, control parameter range, control lag, dead time, cycling.

A.3 Unit III : Control modes

Discontinuous- two positions, multi position, floating control, introduction to continuous control modes- proportional, integral, derivative composite modes.

A.4 Unit IV : PLC Basics

Programmable Controllers - functional diagram, operation, programming, PLC system, I/O modules and interfacing, CPU processor, programming equipment, programming formats, construction of PLC ladder diagrams, devices connected to I/O modules.

B Process Control Instrumentation Lab

- (a) Measure the water level using air-bubble method and calibrate it.
- (b) Calibrate the turbine flow sensors used in the process control board.
- (c) Study the characteristics of an ON-OFF controller with error band at different set points.

C Suggested books

- (a) Curtis Johnson, Process Control Instrumentation Technology.
- (b) Donald P Eckman, Automatic Process Control.
- (c) Katsuhiko Ogata, Modern Control Engineering

13

INS-352 (INS-HC-3016)

Biomedical Instrumentation

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- A paper on Sensors and Transducers is desirable.

Course Outcomes

- CO1: Students will gain an understanding of the fundamental concepts and principles behind bioelectric potentials and their measurement.
- CO2: Students will be able to discuss cardiovascular system and related issues.
- CO3: Students will be able to explain the fundamental concepts and principles behind ultrasound imaging and its applications.
- CO4: Students will be able to explain the magnetic resonance imaging systems.

A Theory

A.1 Unit I : Introduction

Introduction to bioelectric potential, bio-amplifier, components of man Instrument system, design factors of biomedical instruments, types of bio potential electrodes.

A.2 Unit II : Cardio Vascular System

Origin of (Electrocardiography) ECG signals, Instruments of ECG, bipolar system lead system I, II, III, Eithoven triangle, Augmented lead system, unipolar chest lead system, types of display. Respiratory system: Types of volume, types of measurements, Instrumentations of respiratory system, pneu

A.3 Unit III : Medical Imaging System

Ultra sound, properties, beam width, its generation detection, types of transducers, diagnostic application - A Scan, B Scan, M Scan. Radiography- conventional X ray, properties, generation of X-ray, X ray computed tomography (CT scanner) and computer-aided tomography (CAT).

A.4 Unit IV : *Magnetic Resonance Imaging*

Principles of NMR, NMR Imaging system - Image reconstruction Techniques - Basic NMR components - Biological efforts of NMR Imaging - Advantages of NMR Imaging System. Principles of EMR/EPR/ESR. Biomedical EMR.

A.5 Unit IV : *Advanced Bio Medical Systems*

Pacemakers- Need for Cardiac pacemaker - External Pace makers - Implantable Pace makers - Pacing system Analyzer - Defibrillators - Need for a Defibrillator - DC Defibrillator - Implantable Defibrillators - Pacer - Cardioverter - Defibrillator Analyzers - Physio therapy and electro-therapy equipment- High frequency heat therapy - short wave diathermy- microwave and ultrasonic therapy - pain relief through electrical stimulation.

B Biomedical Instrumentation Lab

- (a) Characterisation of bio potential amplifier for ECG signals.
- (b) Study on ECG simulator.
- (c) Recording of EEG
- (d) Measurement of heart sound using electronic stethoscope.
- (e) Study of pulse rate monitor with alarm system.
- (f) Determination pulmonary function.
- (g) Study on ultrasound transducers based on medical system.

C Suggested books

- (a) Cromwell L., Wiebell F. J., Pfeiffer EA, Biomedical Instrumentation and Measurements, Second edition, Prentice Hall.
- (b) Carr J. J, Brown J. M. Introduction to Biomedical Equipment Technology, Fourth edition, Pearson Education Inc.
- (c) Khandpur R.S., Handbook of Biomedical Instrumentation, Second edition, Tata McGraw-Hill Publishin..
- (d) Joseph D. Bronzino, The Biomedical Engineering Handbook, IEEE Press.
- (e) J S Webster, Medical Instrumentation-Application and Design
- (f) Richard Aston, Principles of Biomedical Instrumentation & Measurement, Merrill Publishing Company.

14

INS-353 (INS-HG-4036)

Machine Intelligence

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Basic programming knowledge in C and Python / Matlab

Course Outcomes

- CO1: Understand the components of AI, differences between human and machine intelligence, and structured knowledge representation using semantic networks, frames, and expert system architecture.
- CO2: Develop the ability to apply supervised and unsupervised learning techniques using linear and logistic regression, decision trees, k-nearest neighbors, and clustering algorithms.
- CO3: Gain knowledge on the history and development of neural networks, types of activation functions, architectures of fully connected and layered neural networks, and training and validation techniques using backpropagation algorithm.
- CO4: Acquire knowledge on fuzzy logic operations on fuzzy sets, temperature control system using fuzzy logic, and introduction to genetic algorithms including reproduction, cross-over, mutation, scaling, and fitness.

A Theory

A.1 Unit I : *Introduction*

Components of AI, human intelligence vs. machine intelligence, Knowledge Acquisition, Representation and organisation: Structured Knowledge representation using Semantic Networks, Frames, Expert system architecture, functions of various parts, Mechanism and role of inference engine, Types of expert systems.

A.2 Unit II : *Machine Learning*

Supervised Learning: Concept of labelled data, training data, and testing data. Linear regression, logistic regression, decision trees, and k-nearest neighbours. Unsupervised Learning: Concept of unlabelled data, clustering, k-means, and principal component analysis (PCA). Evaluation Metrics: Confusion matrix, accuracy, precision, recall, F1 score, and ROC curve. Cross Validation and tuning of hyper-parameters.

A.3 Unit III : Artificial Neural Networks

History and development of neural networks. Introduction to perceptrons, Perceptron learning algorithm, Structure and function of a single neuron, artificial neuron models, Types of activation functions, Neural network architectures: Fully connected, layered, acyclic, feed forward. Back-propagation algorithm, training and validation techniques. Application areas of neural networks.

A.4 Unit IV : Fuzzy Logic & Genetic algorithms

Fuzziness vs. probability, Crisp logic vs. fuzzy logic, Fuzzy sets and systems, operations on sets, fuzzy relations, membership functions, fuzzification interface, knowledge/rule base, decision making logic, defuzzification interface, Applications of Fuzzy Logic in process Control and motion control. Introduction to Genetic algorithms. Reproduction, cross-over, mutation, scaling, and fitness of Genetic algorithms.

B Machine Intelligence Lab

Implement programs using Python or Matlab Fuzzy logic and Neural Network toolbox exemplifying

- (a) Implement a linear regression model and train it on a given dataset. Evaluate the model's performance using various metrics like mean squared error and R-squared.
- (b) Implement a logistic regression model and train it on a given dataset. Evaluate the model's performance using various metrics like accuracy, precision, recall, and F1 score.
- (c) Implement a k-Nearest Neighbors (k-NN) algorithm and use it for classification on a given dataset. Evaluate the model's performance using various metrics like accuracy, precision, recall, and F1 score.
- (d) Implement a decision tree algorithm and use it for classification on a given dataset. Evaluate the model's performance using various metrics like accuracy, precision, recall, and F1 score.
- (e) Implement a multi-layer perceptron (MLP) neural network. Train the network on a given dataset and evaluate its performance using various metrics like accuracy, precision, recall, and F1 score.
- (f) Implement basic fuzzy logic operations (AND, OR, NOT)
- (g) Implement fuzzy logic operations for linguistic variables (e.g. age, temperature, etc.)
- (h) Design a fuzzy logic based temperature control system using Python.
- (i) Implement a simple temperature sensor using a microcontroller and Python.

C Suggested books

- (a) Gareth James, Daniela Witten, Trevor Hastie, Rob Tibshirani, *An Introduction to Statistical Learning*, Springer. Free eBook available at <https://www.statlearning.com/>
- (b) Timothy J. Ross, *Fuzzy logic with Engineering Applications*, McGraw Hill.
- (c) S. Rajasekaran, G. A. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic And Genetic Algorithm: Synthesis and Applications*, PHI Learning Pvt. Ltd.
- (d) Martin T. Hagan, Howard B. Demuth, Mark H. Beale, *Neural Network Design*, PWS Publishing Company, Thomson Learning.

15

INS-354 (INS-SE-4024)

Virtual Instrumentation

Total Credits : 4 (Theory : 03, Lab : 01)

Hours : 75 (Theory : 45, Lab Contact : 30)

Prerequisites:

- Good knowledge on analog and digital electronics
- Basic knowledge on control system
- Basic knowledge on programming techniques

Course Outcomes

- CO1: Describe Virtual instrumentation, its architecture, types and advantages.
- CO2: Discuss implementation of LabVIEW in VI design.
- CO3: Discuss LabVIEW programming techniques.
- CO4: Explain the implementation strategy of Different simulation software.

A Theory

A.1 Unit I : Introduction

Evolution of Virtual Instruments, architecture of virtual Instruments, Definition, Distributed virtual instrumentation, Virtual instruments vs Traditional instruments, Advantages of VI, Evolution of LabVIEW, Creating VI using LabVIEW.

A.2 Unit II : The LabVIEW Programming Environment

Controls/ Indicators, Simple programming structures and Timing Issues, Basic operations, controls and indicators.

A.3 Unit III : Programming Techniques

VIs and sub-VIs, Debugging a VI and Sub-VIs, loops & charts, arrays, clusters, graphs, case & sequence structures, formula modes, local and global variable, string & file input, Graphical programming in data flow.

A.4 Unit IV : *Introduction to Simulation*

Introduction to circuit simulation (Multisim or scilab G2 for basic circuit simulation- transistor and opamp), Introduction to Design, analyze, and test system and software architectures (Simulink or Xcos (Scilab) for control system- basic block diagram), Introduction to MATLAB/ SCILAB coding techniques.

B Virtual Instrumentation Lab

- (a) Find whether the given number is odd or even.
- (b) Develop a VI to see sin wave and cosine wave in the same graphical plot.
- (c) Write a LabVIEW programme to find the sum of first 10 natural numbers using while loop.
- (d) Write a LabVIEW programme to find the factorial of a given number using while loop.
- (e) Write a LabVIEW programme to find the sum of first 10 natural numbers using for loop.
- (f) Write a LabVIEW programme to find the factorial of a given number using for loop.

C Suggested books

- (a) S. Sumathi and P. Surekha, LabVIEW based Advanced Instrumentation Systems.
- (b) John Essick, Hands on Introduction to LabVIEW for Scientists and Engineers.
- (c) S. Gupta, J.P. Gupta, PC Interfacing for Data Acquisition and Process Control, ISA.
- (d) Gary Johnson, LABVIEW Graphical Programming, McGraw Hill.
- (e) Technical Manuals for DAS Modules of Advantech and National Instruments. L.T. Amy, Automation System for Control and Data Acquisition, ISA.
- (f) For MATLAB/SIMULINK/MULTISIM visit <https://in.mathworks.com/>