List of New Course(s) Introduced

Department : *Information Technology Engineering*

Programme Name : *B.Tech.*

Academic Year : 2021-22

List of New Course(s) Introduced

Sr. No.	Course Code	Name of the Course
01.	IT203TPC01	ANALOG ELECTRONICS CIRCUITS
02.	IT203TPC02	DIGITAL ELECTRONICS
03.	IT203TPC03	OBJECT ORIENTED PROGRAMMING
04.	IT203PES06	ANALOG ELECTRONIC CIRCUITS LAB
05.	IT203PPC01	DATA STRUCTURE LAB
06.	IT203PPC02	DIGITAL ELECTRONICS LAB
07.	IT203PPC03	OBJECT ORIENTED PROGRAMMING LAB
08.	IT204TPC03	MANAGEMENT-I
09.	IT204PPC01	COMPUTER ORGANISATION & ARCHITECTURE LAB
10.	IT204THS02	OPERATING SYSTEM LAB
11.	IT204PPC03	IT WORKSHOP

Minutes of Meetings (MoM) of Board of Studies (BoS)

Academic Year : 2021-22

School : *School of Studies of Engineering and Technology*

Department : *Information Technology Engineering*

Date and Time : *September 01, 2021 – 04.00 PM*

Venue : *Department of Information Technology*

Minutes of Meeting

Minutes of Meeting Dated 01/09/2021

A Meeting of BoS in Information Technology was held today on 01/09/2021 at 04.00 PM. The Following Members have attended the meeting.

1. Dr. Rohit Raja, BoS Chairman, Dept. of Information Technology, SoS-E&T, GGV.
2. Prof. Sunita Agrawal, Professor, MNIT Allabhabad
3. Mr. Ashish Shrivastava, SDO, BSNL, Bilaspur
4. Mr. Agnivesh Pandey, Member, BoS, Dept. of IT, SoS-E&T, GGV
5. Dr. Amit Kumar Khaskalam, Invited Member
6. Dr. Rajesh Mahule, Invited Member
7. Dr. Santosh Soni, Invited Member
8. Mr. Abhishek Jain, Invited Member
9. Mr. Pankaj Chandra, Invited Member
10. Mr. Deepak Kant Netam, Invited Member
11. Mr. Suhel Ahamed, Invited Member
12. Mr. Amit Kumar Dewangan, Invited Member.
13. Mrs. Aradhana Soni, Invited Member.

The Head of Department welcomed all members of BoS in the meeting and then the following agenda was discussed in the meeting.

1. The Scheme and Syllabus of B.Tech. IT – 2nd Year 2021-22 (3rd and 4th Semester) CBCS has been discussed and approved.

The committee discussed and approved the scheme and syllabi. The following courses were revised in the B. Tech. Second year (III and IV Semesters):

- MATHEMATICS-III (IT203TBS05)
- DATA STRUCTURE AND ALGORITHMS (IT203TPC01)
- DISCRETE MATHEMATICS (IT204TPC01)
- OPERATING SYSTEMS (IT204TPC03)
- COMPUTER ORGANISATION & ARCHITECTURE (IT204TPC02)
- DESIGN & ANALYSIS OF ALGORITHMS (IT204TPC04)

HEAD
2015/22
HEAD
Department of Information Technology
Institute of Technology
Guru Ghasidas Vishwavidyalaya, Bilaspur
(Central University)

The following new courses were introduced in the B. Tech. Second year (III and IV Semesters):

- DIGITAL ELECTRONICS (IT203TPC02)
- OBJECT ORIENTED PROGRAMMING (IT203TPC03)
- ANALOG ELECTRONICS CIRCUITS (IT203TPC01)
- OBJECT ORIENTED PROGRAMMING LAB (IT203PPC03)
- ANALOG ELECTRONIC CIRCUITS LAB (IT203PES06)
- DIGITAL ELECTRONICS LAB (IT203PPC02)
- DATA STRUCTURE LAB (IT203PPC01)
- MANAGEMENT-I (IT204TPC03)
- OPERATING SYSTEM LAB (IT204THS02)
- COMPUTER ORGANISATION & ARCHITECTURE LAB (IT204PPC01)
- IT WORKSHOP (IT204PPC03)

HEAD
20/5/22
HEAD
Department of Information Technology
Institute of Technology
Guru Ghasidas Vishwavidyalaya, Bilaspur
(Central University)

Scheme and Syllabus

SCHEME FOR EXAMINATION
B.TECH (FOUR YEAR) DEGREE COURSE
SECOND YEAR, INFORMATION TECHNOLOGY
SEMESTER III
EFFECTIVE FROM SESSION 2021-22

SL. NO.	SUBJECT CODE	SUBJECTS	PERIODS/ WEEK			EVALUATION SCHEME			CREDITS
			L	T	P	IA	ESE	TOTAL	
THEORY									
1	IT203TES06	ANALOG ELECTRONIC CIRCUITS	3	0	0	30	70	100	3
2	IT203TPC01	DATA STRUCTURE & ALGORITHMS	3	0	0	30	70	100	3
3	IT203TPC02	DIGITAL ELECTRONICS	3	0	0	30	70	100	3
4	IT203TBS05	MATHEMATICS-III	3	1	0	30	70	100	4
5	IT203TPC03	OBJECT ORIENTED PROGRAMMING	3	1	0	30	70	100	4
PRACTICAL									
1	IT203PES06	ANALOG ELECTRONIC CIRCUITS LAB	0	0	4	30	20	50	2
2	IT203PPC01	DATA STRUCTURE LAB	0	0	4	30	20	50	2
3	IT203PPC02	DIGITAL ELECTRONICS LAB	0	0	4	30	20	50	2
4	IT203PPC03	OBJECT ORIENTED PROGRAMMING LAB		0	4	30	20	50	2
TOTAL CREDITS									25
IA- INTERNAL ASSESSMENT, ESE-END SEMESTER EXAMINATION, L-LECTURE, T-TUTORIAL, P-PRACTICAL									

SCHEME FOR EXAMINATION
B.TECH (FOUR YEAR) DEGREE COURSE
SECOND YEAR, INFORMATION TECHNOLOGY
SEMESTER IV
EFFECTIVE FROM SESSION 2021-22

SL. NO.	SUBJECT CODE	SUBJECTS	PERIODS/ WEEK			EVALUATION SCHEME			CREDITS
			L	T	P	IA	ESE	TOTAL	
THEORY									
1	IT204TPC01	DISCRETE MATHEMATICS	3	1	0	30	70	100	4
2	IT204TPC02	COMPUTER ORGANIZATION & ARCHITECTURE	3	0	0	30	70	100	3
3	IT204TPC03	OPERATING SYSTEMS	3	0	0	30	70	100	3
4	IT204TPC04	DESIGN & ANALYSIS OF ALGORITHMS	3	0	0	30	70	100	3
5	IT204THS02	MANAGEMENT I – MANAGEMENT PROCESS AND ORGANIZATIONAL BEHAVIOUR	3	0	0	30	70	100	3
PRACTICAL									
1	IT204PPC01	COMPUTER ORGANIZATION & ARCHITECTURE LAB	0	0	4	30	20	50	2
2	IT204PPC02	OPERATING SYSTEMS LAB	0	0	4	30	20	50	2
3	IT204PPC03	IT WORKSHOP	1	0	2	30	20	50	2
TOTAL CREDITS									22
IA- INTERNAL ASSESSMENT, ESE-END SEMESTER EXAMINATION, L-LECTURE, T-TUTORIAL, P-PRACTICAL									

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203TES06	3	0	0	3 HOURS	30	70	3

ANALOG ELECTRONIC CIRCUITS

Course Objective

1. To apply concepts for the design of low frequency Amplifiers
2. To apply concepts for the design of high frequency Amplifiers
3. To analyze the effects of negative feedback on amplifier circuits.
4. To analyze and determine the different oscillator circuits for waveform Generation
5. To apply concept of the operation of various types of power amplifier circuits.
6. To apply concept of Millers Theorem.

UNIT- I

Low frequency transistor, amplifier, graphical analysis of CE Amplifier, h-parameter models for CB, CE, CC configurations and their interrelationship, analysis and comparison of the three configurations, linear analysis of transistor circuits, Miller's Theorem: Cascading, simplified models and calculation of CE and CC Amplifiers, effect of emitter resistance in CE amplifiers, cascade amplifiers, Darlington pair, analysis of single stage FET amplifiers-CS and CD configuration.

UNIT II

High frequency transistor amplifier, CE hybrid pi model, validity and parameter variation, current gain with resistive load, frequency response of a single stage CE amplifier, gain bandwidth product, CC stage high frequencies, multistage amplifier, classification, distortion in amplifiers, frequency response, bode plots, step response, pass band of cascaded stages, response of a two stage RC coupled amplifier at low and high frequencies, sources of noise in transistor circuits, noise figure.

UNIT III

Feedback Amplifiers: Classification, feedback concept, ideal feedback amplifier, properties of negative feedback amplifier topologies: method of analysis of feedback amplifier, voltage series feedback, voltage series feedback pair, current series, current shunt, voltage shunt feedback, effect of feedback on amplifier bandwidth and stability.

UNIT IV

Large Signal/power amplifier, classification, large signal amplifier characteristics, class A amplifiers, class A amplifier with direct coupled resistive load, transformer coupled class A amplifier, class A push pull amplifiers, class B amplifiers, transformer coupled push pull class B amplifier, complementary symmetry push pull class B amplifier, class AB amplifier, class C amplifier, Harmonic Distortion, Push Pull Amplifiers, Cross over Distortion.

UNIT V

Oscillator: Sinusoidal oscillator, phase shift oscillator, Wien bridge oscillator, Resonant circuit oscillator, LC Colpit, LC Hartley, Amplitude, Frequency, and phase stability analysis of all oscillators. General form of oscillator configuration, crystal oscillator, tuned Amplifiers, classification of tuned

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203TPC02	3	0	0	3 HOURS	30	70	3

DIGITAL ELECTRONICS

Course Objectives:

1. To understand the basic knowledge of digital logic and components.
2. Design of combinational circuits and sequential circuits.
3. Application of knowledge to understand digital electronics circuits.
4. To impart how to design Digital Circuits.

Course Outcome (COs):

At the end of this course, students will demonstrate the ability to

- Convert different type of codes and number systems which are used in digital communication and computer systems.
- Employ the codes and number systems converting circuits and Compare different types of logic families which are the basic unit of different types of logic gates in the domain of economy, performance and efficiency.
- Analyze different types of digital electronic circuit using various mapping and logical tools and know the techniques to prepare the most simplified circuit using various mapping and mathematical methods.
- Design different types of with and without memory element digital electronic circuits for particular operation, within the realm of economic, performance, efficiency, user friendly and environmental constraints.
- Apply the fundamental knowledge of analog and digital electronics to get different types analog to digitalized signal and vice-versa converters in real world with different changing circumstances.
- Assess the nomenclature and technology in the area of memory devices and apply the memory devices in different types of digital circuits for real world application.

UNIT 1 - Fundamentals of Digital systems and logic families

Digital signals, digital circuits, AND,OR, NOT, NAND, NOR and Exclusive OR operations, Boolean algebra, examples of IC gates, number systems- binary, signed binary, octal, Hexadecimal number, binary arithmetic, Once's and two's complements, arithmetic codes, error detecting, and correcting codes, characteristics of digital ICs, digital logic families, TTL, schottky TTL and CMOS logic , interfacing CMOS and TTL, Tri-state logic.

UNIT 2 - Combinational Digital Circuits

Standard representation for logic function, K map representation, simplification of logic functions, using K map, minimization of logical functions. Don't care conditions, Multiplexes, De- Multiplexes, / Decoders, Adders, Sub tractors, BCD arithmetic, carry look ahead, serial adders , ALU , elementary ALU design, popular MSI chips , digital comparator , parity checker, / generator, code converters, priority encoders, decoders/ drivers, for display devices, Q-M method of function realization.

UNIT 3 - Sequential circuits and systems

A 1 bit memory, the circuits properties, of Bi-stable latch, the clocked SR flip flop, JK flip flops, T flip flops, D flip flops, applications of flip flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters, counter's design using flip flops, special counter IC's, Asynchronous sequential counters, applications of counters.

UNIT 4 – A/D and D/A converters

Digital to analog converters: weighted registers/ converters, R-2R Ladder, D/A converters, specifications for D /A converters, examples of D /A converter ICs, sample and hold circuits, Analog to digital converters: quantization and encoding, parallel comparator, A/D converter, successive approximation A/D converter, counting A/D converter, dual slop A / D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs.

UNIT 5 - Semiconductor memories and Programmable logic devices

Memory organization and operation, expanding memory size, classification and characteristics of memories, Sequential memories, read-only memory(ROM), read and write memory (RAM), content addressable memory (CAM) , charge coupled device memory (CCD) , commonly used memory chips, ROM as a PLD, Programmable logic array, Programmable array logic, complex Programmable logic devices (CPLDS) , Field Programmable Gate Array (GPGA).

Text / References:

1. M M Mano, "Digital logic and Computer design", Pearson Education India.
2. R P. Jain, " Modern Digital Electronics" , McGraw Hill Education.
3. A kumar, "Fundamentals of Digital Circuits ", Prentice Hall India.
4. S Salivahanan and S Arivazhagan " Digital Circuits and Design" OXFORD University Press.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203TPC03	3	1	0	3 HOURS	30	70	4

Object Oriented Programming

Course Objectives:

1. To understand and Practice Programming Construct: Variable, Operators, Control Structures, Loop, Functions, learn the concept of class and object and develop classes for simple applications with C++.
2. To learn how to implement Constructors, copy constructors and destructor functions.
3. To learn how to overload functions and operators in C++.
4. To learn how to design C++ classes for code reuse and perform inheritance.
5. To learn working with files and handle exceptions in program.

UNIT I

Overview of C++ : Object oriented programming, Concepts, Advantages, Usage. C++ Environment: Program development environment, the language and the C++ language standards. Introduction to various C++ compilers, C++ standard libraries, Prototype of main() function, Data types. C++ as a superset of C, New style comments, main function in C++, meaning of empty argument list, function prototyping, default arguments and argument matching.

User defined data types: enumerated types, use of tag names, anonymous unions, scope of tag names Classes & Objects : Classes, Structure & Classes, Inline Function, Scope Resolution operator, Static Class Members: Static Data Member, Static Member Function, Passing Objects to Function, Returning Objects, Object Assignment. Friend Function, Friend Classes

UNIT II

Array, Pointers References & The Dynamic Allocation Operators: Array of Objects, Pointers to Object, Type Checking C++ Pointers, The This Pointer, Pointer to Derived Types, Pointer to Class Members, References: Reference Parameter, call by reference and return by reference Passing References to Objects, Returning Reference, Independent Reference, C++'S Dynamic Allocation Operators, Initializing Allocated Memory, Allocating Array, Allocating Objects.

Constructor & Destructor: Introduction, Constructor, access specifies for constructors, and instantiation, Parameterized Constructor, Multiple Constructor in A Class, Constructor with Default Argument, Copy Constructor, Destructor.

UNIT III

Overloading as polymorphism: Function & Operator Overloading : Function Overloading, Overloading Constructor Function Finding the Address of an Overloaded Function, Operator Overloading: Creating A Member Operator Function, Creating Prefix & Postfix Forms of the Increment & Decrement Operation, Overloading The Shorthand Operation (i.e., +=, -= etc), Operator Overloading Restrictions, Operator Overloading Using Friend Function, Overloading Some Special Operators like [ ], ( ), -, Comma Operator, Overloading << etc.

UNIT IV

Inheritance : Base Class Access Control, Inheritance & Protected Members, Protected Base Class Inheritance, Inheriting Multiple Base Classes, Constructors, Destructors & Inheritance, When

Constructor & Destructor Function are Executed, Passing Parameters to Base Class Constructors, Granting Access, Virtual Base Classes .
Virtual Functions & Polymorphism: Virtual Function, Pure Virtual Functions, Early Vs. Late Binding.

UNIT V

Working with files: File & stream, Opening and closing a file, read ( ) and write ( ) functions, detecting end of file.
Templates and Exception Handling: Exception handling in C++, try, throw, catch sequence, multiple catch blocks, uncaught exceptions, catch-all exception handler

Course Outcomes:-

1. Understand the C++ language features. Use the control structure and data types in C++. Write simple programs using classes and objects.
2. Understand the concepts of arrays, pointers, references and use of dynamic allocation operators. Write simple programs to implement Constructor & destructor concepts.
3. Understand the concept of Operator overloading and type conversion.
4. Understand the concepts of inheritance and virtual functions.
5. Understand file handling concepts, generic class and I/O exception handling.

Reference Books:

Object Oriented Programming with C++ by M. P. Bhave, S. A. Patekar, Pearson Education

Object Oriented Programming With C++ by E. Balaguruswamy.

Object Oriented Programming in turbo C++ by Robert Lafore.

Programming with C++ by D. Ravichandan.

Programming with C++ (SOS) by Hubbard.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203PES06			4	3 HOURS	30	20	2

Course Objective

- Understand the circuit configurations and connectivity of Amplifiers and Study of frequency response
- Design and test of analog circuits using OPAMPs
- Understand the feedback configurations of transistor and OPAMP circuits
- Use of circuit simulation for the analysis of electronic circuits.

List of Experiment

1. **RC coupled amplifier**
2. **Darlington Emitter Follower**
3. **Voltage Series Feedback Amplifier**
4. **RC Phase shift Oscillator**
5. **Hartley & Colpitt's Oscillator**
6. **Clipping circuits**
7. **Clamping circuits**
8. **Op-Amp applications**
9. **ZCD & Schmitt trigger**
10. **Full wave Precision Rectifier**
11. **Voltage Regulator**
12. **Digital-Analog Converter**
13. **Analog-Digital Converter**

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT203PPC01	0	0	4	4 HOURS	30	20	2

DATA STURCTURE LAB

Course Objective

The course is designed to develop skills to design and analyze simple linear and nonlinear data structures. It strengthens the ability to the students to identify and apply the suitable data structure for the given real-world problem. It enables them to gain knowledge in practical applications of data structures.

Course Outcome

At the end of this lab session, the student will

1. Design programs using a variety of data structures such as Stacks, Queues, Array, Binary Trees, and Linked List.
2. Analyze and implement various kinds of searching and sorting techniques.
3. Have practical knowledge on the applications of data structures.

List of Practical's

1. Write a C program for declaration, assignment, and accessing the arrays elements.
2. Write a C Program to Find Average Marks obtained by a class of 30 Students in a Test.
3. Write a C program to perform Array Insertion Operation.
4. Write a C program to perform Array Deletion Operation.
5. Write a C program to implement Linear Search.
6. Write a C program to implement Binary Search.
7. Write a C program to implement Bubble Sort.
8. Write a C program to implement Merging operation.
9. Write a program in C to create and display Singly Linked List.
10. Write a program in C to create a singly linked list of n nodes and count the number of nodes.
11. Write a program in C to insert a new node at the beginning of a Singly Linked List.
12. Write a program in C to insert a new node at the end of a Singly Linked List.
13. Write a program in C to insert a new node after a given location of Singly Linked List.
14. Write a program in C to delete first node of Singly Linked List.
15. Write a program in C to delete the last node of Singly Linked List.
16. Write a program in C to delete a node from the middle of Singly Linked List.
17. Write a program in C to search an existing element in a singly linked list.
18. Write C programs to implement the stack push operation using an array.
19. Write C programs to implement the stack pop operation using an array.

References books:

10. Lipschutz, "Data Structures with C" Schaum's Outline Series, TMH.
11. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
IT203PPC02	0	0	4	4 HOURS	30	20	2

DIGITAL ELECTRONICS LAB

Course Objectives

The objectives are to study

1. To provide students basic experimental experiences in constructing digital circuits, measuring the experimental data and analysis of the results.
2. To develop skills to design various combinational and sequential circuits using electronics devices.

Course Outcomes (COs)

After studying this course the students would gain enough knowledge.

1. To acquire knowledge about electronic components and hardware devices required for designing digital electronics circuits.
2. Foster ability to identify, analyze and design combinational circuits.
3. Foster ability to design various synchronous and asynchronous sequential circuits.
4. To acquire knowledge about internal circuitry and logic behind any digital system.
5. To develop skill to build, and troubleshoot digital circuits.

List of Experiments

1. To implement Logic gates using ICs (7400, 7402, 7404, 7408, 7410, 7411, 7420, 7427, 7432, 7486).
2. Implementation of Combinational Circuits.
3. To verify NAND and NOR gates are universal gates.
4. Implementation of Combinational Logic Design using 74** ICs.
5. Simplification of Boolean expression using Karnaugh Map Method.
6. To implement Adder and Subtractor circuits:- (Half and Full using simple gates and universal gates).
7. To verify the truth table of Binary (2 bit) to decimal decoder and octal to decimal decoder.
8. Functional table verification of Latches (i) SR-Latch with NOR Gates (ii) SR-Latch with NAND Gates.

SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
1T203PPC03			4	3 HOURS	30	20	2

Object Oriented Programming with C++ Lab

Course Objectives:

1. To understand and Practice Programming Construct: Variable, Operators, Control Structures, Loop, Functions, learn the concept of class and object and develop classes for simple applications with C++.
2. To learn how to implement Constructors, copy constructors and destructor functions.
3. To learn how to overload functions and operators in C++.
4. To learn how to design C++ classes for code reuse and perform inheritance.
5. To learn working with files and handle exceptions in program.

List of Experiments:-

1. Write a program to display message using cout statement.
2. Write a program to calculate average of five numbers given by user.
3. Write a program to calculate compound interest given P, R and T.
4. Write a program to calculate factorial of a given number.
5. Write a program to generate n numbers of fibonacii series. Value of n should be provided by user.
6. Write a function to calculate the power of a number raised to another number using function. Write appropriate main() function to read and display the result.
7. Write a function factorial to calculate the factorial of a number, write appropriate main function also.
8. Write a function swap to swap the value of two integer variables. Write appropriate main function for the program.
9. Write a function to perform sorting using bubble sort algorithm. Use arrays to store the list of numbers. Also write main() function to read contents and display output.
10. Write a program to perform overloading of area function.
11. Write a program with overloaded volume function. Use volume function to calculate the volume of a cube, cone, sphere etc.
12. Write a program to calculate simple interest. Use default argument for rate. Write main function to exhibit the use of default argument.
13. Write a program to show the use of return by reference.
14. Write a program with at least one function made as inline.
15. Create a structure data type with data items roll number, name, and total marks. Write main function to read data for two students and also display the stored data.
16. Create a class named 'Student' with a string variable 'name' and an integer variable 'roll no'. Assign the value of roll no as '2' and that of name as "John" by creating an object of the class Student.
17. Write a program to print the area of a rectangle by creating a class named 'Area' having two functions. First function named as 'readData' takes the length and breadth of the rectangle as

Aradhana Soni

parameters and the second function named as 'calculateArea' returns the area of the rectangle. Length and breadth of the rectangle are entered through keyboard.

18. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

Name	Year of joining	Address
Ramesh	1994	64-C New Delhi
Sam	2000	68-D Bilaspur
John	1999	26-B-Banglore

19. Define a class to represent a bank account. Include the following members:

Data members:

1. Name of the depositor.
2. Account number.
3. Type of account.
4. Balance amount in the account.

Member functions:

1. To assign initial values.
2. To deposit an amount.
3. To withdraw an amount after checking the balance.
4. To display the name and balance.

Write a main program to test the program.

20. Define a class to represent a bank account (FOR 100 CUSTOMERS). Include the following members:

Data members:

1. Name of the depositor.
2. Account number.
3. Type of account.
4. Balance amount in the account.

Member functions:

1. To assign initial values.
2. To deposit an amount.
3. To withdraw an amount after checking the balance.
4. To display the name and balance.

Write a main program to test the program.

21. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use friend function to carry out the addition operation. The object that stores result may be a DM object or DB object depending on the units in which result is required.

22. Create a class time to store time in hours and minutes. Write a program that can read values for the class objects and add one object with another object storing the result in third object. Use object as arguments to sum() function and object return type from sum() function to assign the sum to the third object.

Ashutosh Soni

23. Write a program to demonstrate the use of Static Keyword. Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of the Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating the object of the Student class. Use "new" operator to create an instance of "name" member of class.
24. Write a program to add two complex number using a friend function. Use appropriate constructor function to initialize the object.
25. Write a program to initialize an object with another using copy constructor.
26. Write a program to show the highest scorer in a test out of three students who appeared in a exam. Use this pointer to refer to objects.
27. Write a program to show the highest scorer in a test out of three students who appeared in a exam. Use this pointer to refer to objects. (USER INPUT)
28. Write a program to illustrate the creation and destruction of objects.
29. Write a program to illustrate pointer to member and pointer to object concepts of 00P.
30. Create two objects of a class with two integer type members. Compare the two operators have same member values. Using overloaded = operator for comparison.
31. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.
32. Write a program to overload << and >> operators to display time object.
33. Write a program to overload <=, >== and == operator to compare time objects.
34. Write a program to perform overloading of function call operator.
35. Write a program to perform overloading of subscripting operator.
36. Write a Program to Concatenate two Strings Using Operator Overloading(+).
37. Write a program to convert time type object to an Integer value and integer type value to object of time type. Use appropriate data type conversion functions to perform the required conversion.
38. Write a program for an inventory of product in store. Use appropriate data type conversion functions to perform the required conversion.
39. Define a class Student with data members as rollno and name. Derive a class Fees from student that has a data member fees and functions to submit fees and generate receipt.. Derive another class Result from Student and display the marks and grade obtained by the student.

Course Outcomes:-

1. Understand the C++ language features. Use the control structure and data types in C++. Write simple programs using classes and objects.
2. Understand the concepts of arrays, pointers, references and use of dynamic allocation operators. Write simple programs to implement Constructor & destructor concepts.
3. Understand the concept of Operator overloading and type conversion. Write simple programs using overloaded operators.
4. Understand the concepts of inheritance and virtual functions. Write simple programs to implement inheritance and virtual functions.
5. Understand file handling concepts, generic class and I/O exception handling. Write small programs to implement file handling concepts and exception handling.

Aradhana Soni

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204THS02	3	0	0	3 hours	30	70	3

MANAGEMENT PROCESS AND ORGANIZATIONAL BEHAVIOUR

Course Objectives:

1. To help the students to develop cognizance of the importance of Management processes.
2. To enable students to describe how people behave under different conditions and understand why people behave as they do.
3. To provide the students to analyse specific strategic human resources demands for future action.
4. To enable students to synthesize related information and evaluate options for the most logical and optimal solution such that they would be able to predict and control management processes, human behaviour and improve results.

Course Outcomes (Cos):

On completion of this course, the students will be able to

1. To understand the concept of Management.
2. Demonstrate the applicability of the concept of Management processes to understand the functioning of the organization.
3. Demonstrate the applicability of the concept of organizational behavior to understand the behavior of people in the organization.
4. Analyze the complexities associated with management of the group behavior in the organization.
5. Demonstrate the applicability to manage the organization.

UNIT -I

School of Management Thought: Evolution of Management thought, Systems and Contingency approach of management, Decision Theory School.

UNIT -II

Managerial processes, functions, skills and roles in an organization. Nature, process and technique of planning, Organizing, Staffing, Directing, Coordinating, Control.

UNIT -III

Organizational Behavior: Concept, Significance, Understanding and Managing individual behavior – Personality, Perceptions, Values, Attitudes, Learning, Work-motivation, Individual Decision Making and Problem solving.

UNIT -IV

Understanding and Managing Group Processes: Interpersonal and Group dynamics. Applications of emotional intelligence in organizations. Group decision making. Leadership and Influence Process : Concept, styles and Theories.

UNIT – V

Understanding and Managing Organizational Systems, Organizational Conflict — sources, pattern levels and types of conflict. Organizational design and structure. Work stress.

Suggested Readings

1. Koontz, Harold, Cyril O'Donnell, and Heinz, Whelrich. Essentials of Management. New Delhi: Tata Mc Graw Hill.
2. Robbins, S.P. Organizational Behaviour. New Delhi: PHI.
3. Luthans, F. Organisational Behaviour. NewYork: Mc Graw Hill.

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204PPC01	0	0	4	4 HOURS	30	20	2

COMPUTER ORGANIZATION & ARCHITECTURE LAB

Course Objectives:

The Objective is to expose the students to the various key aspects of Computer Organization & Architecture by enabling them to perform the experiments with support of a design and simulation in Logisim, GNUsim8085 or other simulator.

List of Practical's

1. Write the working of 8085 simulators GNUsim8085 and basic architecture of 8085 along with small introduction.
2. Study the complete instruction set of 8085 and write the instructions in the instruction set of 8085 along with examples.
3. Write an assembly language code in GNUsim8085 to implement data transfer instruction.
4. Write an assembly language code in GNUsim8085 to store numbers in reverse order in memory location.
5. Write an assembly language code in GNUsim8085 to implement arithmetic instruction.
6. Write an assembly language code in GNUsim8085 to add two numbers using lxi instruction.
7. Write an assembly language code in GNUsim8085 to add two 8 bit numbers stored in memory and also storing the carry.
8. Write an assembly language code in GNUsim8085 to find the factorial of a number.
9. Write an assembly language code in GNUsim8085 to implement logical instructions.
10. Write an assembly language code in GNUsim8085 to implement stack and branch instructions.
11. Design Ripple Carry Adder using simulator
12. Design Carry-Look-Ahead Adder using simulator
13. Design Registers and Counters using simulator
14. Design Combinational Multipliers using simulator
15. Booth's Multiplier using simulator
16. Arithmetic Logic Unit using simulator
17. Memory Design using simulator
18. Write a C program to implement Booth's algorithm for multiplication.
19. To write a C program for multiplication of two binary numbers
20. To write C program for sum of two binary number
21. Write a program to implement Division Algorithm.
22. To Study the Logisim Tool
23. Design of Half-adder circuit

Arvindhara Soni

24. Design of Full-adder circuit
25. Design of Ripple -adder circuit

Course Outcomes:

After the course the students are expected to be able to describe in detail the internal and external working of computer.

References:

1. **https://cse.iitkgp.ac.in/~chitta/coldvl/comp.html**

SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
IT204PPC02	0	0	4	4 HOURS	30	20	2

OPERATING SYSTEMS LAB

Course Objectives

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management

List of Experiments

1. **Simulate the following CPU scheduling algorithms.**
   a)FCFS b) SJF c) Round Robin d) Priority.
2. **Write a C program to simulate producer-consumer problem using Semaphores**
3. **Write a C program to simulate the concept of Dining-philosophers problem.**
4. **Write a C program to simulate the following contiguous memory allocation Techniques**
   Worst fit b) Best fit c) First fit.
5. **Simulate all page replacement algorithms a)**
   FIFO b) LRU c) OPTIMAL
6. **Simulate all File Organization Techniques**
   a) Single level directory b) Two leveldirectory
7. **Simulate Bankers Algorithm for Dead Lock Avoidance and dead lock prevention.**
8. **Write a program to simulate disk scheduling algorithms.**
   a) FCFS b) SCAN c) C-SCAN

Course Outcomes

1. Create processes and threads.
2. Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.