



List of Revised Courses

Department	:	Information Technology
Programme Name	:	B.Tech.
Academic Year : 2023-24		

List of Revised Courses

Sr. No.	Course Code	Name of the Course
01.	ITUCLT1	DATA STRUCTURE LAB
02.	ITUCLT2	OBJECT ORIENTED PROGRAMMING WITH C++ LAB
03.	ITUDLT2	OPERATING SYSTEMS LAB



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

Academic Year : 2023-24

School : School of Engineering & Technology

Department : Information Technology

Date and Time : 07-10-2023, 02:30PM

Venue : Smart Class Room G-14 [Hybrid Mode]

Minutes of Meeting Dated 07/10/2023

A Meeting of BoS in Information Technology was held today on 07/10/2023 PM. The Following Members have attended the meeting.

1. Dr. Rohit Raja, BoS Chairman, Dept. of Information Technology, S GGV.
2. Prof. Apurva Desai, Professor, Veer Narmad South Gujarat University
3. Ms. Ashwini Jha, Software Developer, Persistent
4. Mr. Pankaj Chandra, Member, BoS, Dept. of IT, SoS-E&T, GGV.
5. Dr. Santosh Soni, Invited Member
6. Dr. Rajesh Mahule, Invited Member
7. Mr. Abhishek Jain, Invited Member
8. Mr. Agnivesh Pandey, Invited Member
9. Mr. Suhel Ahamed, Invited Member
10. Mr. Deepak Kant Netam, Invited Member
11. Mr. Anand Prakash Rawal, Invited Member
12. Mrs. Akanksha Gupta, Invited Member
13. Dr. Amit Kumar Dewangan, Invited Member
14. Mrs. Aradhana Soni, Invited Member

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

1. The Scheme and Syllabus of B.Tech. IT – 2nd Year NEP (3rd and 4th Sem 2023-24) has been discussed and approved.
2. Scheme and Syllabus of B.Tech. IT – 1st Year (2nd Semester) (it is approved, only one subject proposed by ECE department is to be changed and discussed and approved.
3. The subject code can be changed as per university regulation/ policy time to time.
4. The open elective subjects depend on the syllabus approved by other departments so if there will be any changes that will be incorporated.

The following courses were revised in the B.Tech. IT – 2nd Year (3rd and 4th Sem B.Tech. IT – 2nd Semester:

- ❖ DATA STRUCTURE LAB (ITUCLT1) - B.Tech. 3rd Semester
- ❖ OBJECT ORIENTED PROGRAMMING WITH C++ LAB (ITUCLT2) - B.Tech. 3rd Semester



❖ INTRODUCTION TO ELECTRONICS & COMMUNICATION ENGINEERING (ECUBTE7) – B.Tech. 2nd Semester

The following new courses were introduced in B.Tech. IT – 2nd Year (3rd Semester):

- ❖ SOFTWARE ENGINEERING (ITUCTK2)
- ❖ MULTIMEDIA SYSTEM DESIGN (ITUCTK3)
- ❖ PYTHON FOR DATA SCIENCE (ITUDTT1)
- ❖ PYTHON FOR DATA SCIENCE LAB (ITUDLT1)
- ❖ COMPUTER ORGANIZATION & ARCHITECTURE - INSTITUTE CORE (ITUCO1)
- ❖ DIGITAL SIGNAL PROCESSING (ITUDTK2)
- ❖ COMPUTER APPLICATION IN SOCIAL SCIENCES (ITUDTK3)
- ❖ COMPUTER NETWORK - INSTITUTE CORE (ITUDTO1)
- ❖ FUNDAMENTALS OF PYTHON PROGRAMMING - INSTITUTE CORE (ITUFUN1)
- ❖ MINI PROJECT-I (ITUDPV1)

The meeting ended with a vote of thanks by Head of the Department.

Dr. Rohit Raja
BoS Chairman

(Consent Taken Through Mail)
Prof. Apurva Desai Professor
Veer Narmad South Gujarat University

(Consent Taken Through Mail)
Ms. Ashwini Jha
Software Developer
Persistent

Mr. Pankaj Chandra
Member, BoS

Dr. Santosh Soni
Invited Member

Dr. Rajesh Mahule
Invited Member

Mr. Abhishek Jain
Invited Member

Mr. Agnivesh Pandey
Invited Member

Mr. Suhel Ahamed
Invited Member

Mr. Deepak Kant Netam
Invited Member

Mr. Anand Prakash Rawal
Invited Member

Mrs. Akanksha Gupta
Invited Member

Dr. Amit Kumar Dewangan
Invited Member

Mrs. Aradhana Soni
Invited Member



Scheme and Syllabus

SCHEME FOR EXAMINATION
B.TECH (FOUR YEAR) DEGREE COURSE
SECOND YEAR, INFORMATION TECHNOLOGY
SEMESTER III
EFFECTIVE FROM SESSION 2023-24 (NEP)

SL. NO.	SUBJECT CODE	SUBJECTS	PERIODS/ WEEK			EVALUATION SCHEME			CREDITS
			L	T	P	IA	ESE	TOTAL	
THEORY									
1	ITUCTT1	DATA STRUCTURE & ALGORITHMS	3	0	0	40	60	100	3
2	ITUCTT2	OBJECT ORIENTED PROGRAMMING	3	1	0	40	60	100	4
3	ITUCTT3	DIGITAL ELECTRONICS	3	0	0	40	60	100	3
4	ITUCTE1	MATHEMATICS-III	3	0	0	40	60	100	3
5	ITUCTKX	DEPARTMENT ELECTIVE-I	3	0	0	40	60	100	3
6		INSTITUTE CORE-I	3	0	0	40	60	100	3
PRACTICAL									
1	ITUCLT1	DATA STRUCTURE LAB	0	0	3	25	25	50	1.5
2	ITUCLT2	OBJECT ORIENTED PROGRAMMING WITH C++ LAB	0	0	3	25	25	50	1.5
TOTAL CREDITS									22
IA- INTERNAL ASSESSMENT, ESE-END SEMESTER EXAMINATION, L-LECTURE, T-TUTORIAL, P-PRACTICAL									

LIST OF DEPARTMENT ELECTIVE-I

1	ITUCTK1	COMPUTER ORGANIZATION & ARCHITECTURE
2	ITUCTK2	SOFTWARE ENGINEERING
3	ITUCTK3	MULTIMEDIA SYSTEM DESIGN

LIST OF INSTITUTE CORE-I

1	ITUCTO1	COMPUTER ORGANIZATION & ARCHITECTURE (Not for IT)
2	CSUCTO1	DATA STRUCTURE WITH C++
3	ECUCTO1	DATA COMMUNICATION
4	CEUCTO1	GREEN BUILDINGS
5	CHUCTO1	ENGINEERING MATERIALS
6	MEUCTO1	INTRODUCTION TO THERMODYNAMICS
7	IPUCTO1	I.C. ENGINE



SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
ITUCLT1	0	0	3	3 HOURS	25	25	1.5

DATA STRUCTURE LAB

Course Objective

1. Understand and remember algorithms and its analysis procedure.
2. Introduce the concept of data structures through ADT including List, Stack, Queues.
3. To design and implement various data structure algorithms.
4. To develop application using data structure algorithms.
5. Compute the complexity of various algorithms.

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:

1. Lipschutz, "Data Structures with C" Schaum's Outline Series, TMH.
2. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd.
3. R. Kruse et al, "Data Structures and Program Design in C", Pearson Education Asia.
4. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd.
5. K Loudon, "Mastering Algorithms with C", Shroff Publisher & Distributors Pvt. Ltd.
6. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
7. G A V Pai, "Data Structures and Algorithms", TMH.
8. G.S.Baluja, "Data Structures through C", Dhanpat Rai & Co.
9. Yashavant Kanetkar, "Data Structure Through C", BPB Publication.

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.
4. Design and analyze complexity of different algorithms.
5. Design advance data structure using non linear data structure.



SUB CODE	L	T	P	DURATION	IA	ESE	CREDITS
ITUCLT2	0	0	3	3 HOURS	25	25	1.5

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 fibonacci 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 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.
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 OOP.
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.

Reference Books:

1. Object Oriented Programming with C++ by M. P. Bhawe, S. A. Patekar, Pearson Education
2. Object Oriented Programming With C++ by E. Balaguruswamy.
3. Object Oriented Programming in turbo C++ by Robert Lafore.
4. Programming with C++ by D. Ravichandan.
5. Programming with C++ (SOS) by Hubbard.

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.



**SCHEME FOR EXAMINATION
B.TECH (FOUR YEAR) DEGREE COURSE
SECOND YEAR, INFORMATION TECHNOLOGY
SEMESTER IV
EFFECTIVE FROM SESSION 2023-24 (NEP)**

SL. NO.	SUBJECT CODE	SUBJECTS	PERIODS/ WEEK			EVALUATION SCHEME			CREDITS
			L	T	P	IA	ESE	TOTAL	
THEORY									
1	ITUDTT1	PYTHON FOR DATA SCIENCE	3	1	0	40	60	100	4
2	ITUDTT2	OPERATING SYSTEMS	3	0	0	40	60	100	3
3	ITUDTT3	DISCRETE MATHEMATICS	3	0	0	40	60	100	3
4	ITUDTKX	DEPARTMENT ELECTIVE-II	3	0	0	40	60	100	3
5		INSTITUTE CORE-II	3	0	0	40	60	100	3
PRACTICAL									
1	ITUDLT1	PYTHON FOR DATA SCIENCE LAB	0	0	3	25	25	50	1.5
2	ITUDLT2	OPERATING SYSTEMS LAB	0	0	3	25	25	50	1.5
3	ITUDPV1	MINI PROJECT	0	0	4	50	50	100	2
TOTAL CREDITS									21
IA- INTERNAL ASSESSMENT, ESE-END SEMESTER EXAMINATION, L-LECTURE, T-TUTORIAL, P-PRACTICAL									

LIST OF DEPARTMENT ELECTIVE-II

1	ITUDTK1	DESIGN & ANALYSIS OF ALGORITHMS
2	ITUDTK2	DIGITAL SIGNAL PROCESSING
3	ITUDTK3	COMPUTER APPLICATION IN SOCIAL SCIENCES

LIST OF INSTITUTE CORE-II

1	ITUDTO1	COMPUTER NETWORK (Not for IT)
2	ITUDTO2	FUNDAMENTALS OF PYTHON PROGRAMMING (Not for IT)
3	CSUDTO1	INTRODUCTION TO INFORMATION SCIENCE
4	ECUDTO1	ELECTRONICS DEVICES AND CIRCUITS
5	CEUDTO1	REMOTE SENSING & GIS
6	CHUDTO1	ENERGY AND ENVIRONMENT ENGINEERING
7	ESUDTO1	EFFECTIVE TECHNICAL COMMUNICATION
8	MEUDTO1	INTRODUCTION TO FLUID MECHANICS
9	IPUDTO1	AUTOMOBILE ENGINEERING



SUB CODE	L	T	P	DURATION/WEEK	IA	ESE	CREDITS
ITUDLT2	0	0	3	3 HOURS	25	25	1.5

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-
a) Worst fit b) Best fit c) First fit.
5. Simulate all page replacement algorithms a)
a) FIFO b) LRU c) OPTIMAL
6. Simulate all File Organization Techniques
a) Single level directory b) Two level directory
- 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

Reference Books

1. Milenkovic M. , "Operating System concepts", MGH
2. Tanenbaum A. S. "Operating System design and implementation" , PHI
3. Silberschartz A. and Patterson J.I. , " Operating system concepts", Wisley.
4. Stilling William " Operating System " , Maxwell McMillan International Edition 1992.
5. Dectel H.N. , "An introduction to operating system " , Addison Wisley.

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.
3. Specification of memory organization develops the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.
4. Design and implement file management system.
5. For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.