University of Mumbai

वेबसाईट – mu.ac.in
ईमेल - आयडी - dr.aams@fort.mu.ac.in
aams3@mu.ac.in

विद्याविषयक प्राधिकरणे
सभा आणि सेवा विभाग(ए.ए.एम.एस)
रूम नं. १२८ एम.जी.रोड, फोर्ट,
मुंबई – ४०० ०३२
टेलिफोन नं – ०२२ – ६८३२००३३

(नॅक पुनर्मूल्यांकनाद्वारे ३.६५ (सी.जी.पी.ए.) सह अ++ श्रेणी
विद्यापीठ अनुदान आयोगाद्वारे श्रेणी १ विद्यापीठ दर्जा)

क्र.वि.प्रा.स.से./आयसीडी/२०२५-२६/३७ दिनांक : २७ मे, २०२५

परिपत्रक:-

सर्व प्राचार्य/संचालक, संलग्नित महाविद्यालये/संस्था, विद्यापीठ शैक्षणिक विभागांचे संचालक/ विभाग प्रमुख यांना कळविण्यात येते की, राष्ट्रीय शैक्षणिक धोरण २०२० च्या अमंलबजावणीच्या अनुषंगाने **शैक्षणिक वर्ष २०२५-२६** पासून **पदवी व पदव्युत्तर अभ्यासक्रम** विद्यापरिषदेच्या दिनांक २८ मार्च २०२५ व २० मे, २०२५ च्या बैठकीमध्ये मंजूर झालेले सर्व अभ्यासक्रम मुंबई विद्यापीठाच्या www.mu.ac.in या संकेत स्थळावर NEP २०२० या टॅब वर उपलब्ध करण्यात आलेले आहेत.

(डॉ. प्रसाद कारंडे)
कुलसचिव

मुंबई – ४०० ०३२
२७ मे, २०२५

Scanned with OKEN Scanner

Copy forwarded for information and necessary action to :-	
1	The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Dept)(AEM), dr@eligi.mu.ac.in
2	The Deputy Registrar, Result unit, Vidyanagari drresults@exam.mu.ac.in
3	The Deputy Registrar, Marks and Certificate Unit,. Vidyanagari dr.verification@mu.ac.in
4	The Deputy Registrar, Appointment Unit, Vidyanagari dr.appointment@exam.mu.ac.in
5	The Deputy Registrar, CAP Unit, Vidyanagari cap.exam@mu.ac.in
6	The Deputy Registrar, College Affiliations & Development Department (CAD), deputyregistrar.uni@gmail.com
7	The Deputy Registrar, PRO, Fort, (Publication Section), Pro@mu.ac.in
8	The Deputy Registrar, Executive Authorities Section (EA) eau120@fort.mu.ac.in He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
9	The Deputy Registrar, Research Administration & Promotion Cell (RAPC), rapc@mu.ac.in
10	The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA) dy.registrar.tau.fort.mu.ac.in ar.tau@fort.mu.ac.in
11	The Deputy Registrar, College Teachers Approval Unit (CTA), concolsection@gmail.com
12	The Deputy Registrars, Finance & Accounts Section, fort draccounts@fort.mu.ac.in
13	The Deputy Registrar, Election Section, Fort drelection@election.mu.ac.in
14	The Assistant Registrar, Administrative Sub-Campus Thane, thanesubcampus@mu.ac.in
15	The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan, ar.seask@mu.ac.in
16	The Assistant Registrar, Ratnagiri Sub-centre, Ratnagiri, ratnagirisubcentar@gmail.com
17	The Director, Centre for Distance and Online Education (CDOE), Vidyanagari, director@idol.mu.ac.in
18	Director, Innovation, Incubation and Linkages, Dr. Sachin Laddha pinkumanno@gmail.com
19	Director, Department of Lifelong Learning and Extension (DLLE), dlleuniversityofmumbai@gmail.com

Copy for information :-	
1	P.A to Hon'ble Vice-Chancellor, vice-chancellor@mu.ac.in
2	P.A to Pro-Vice-Chancellor pvc@fort.mu.ac.in
3	P.A to Registrar, registrar@fort.mu.ac.in
4	P.A to all Deans of all Faculties
5	P.A to Finance & Account Officers, (F & A.O), camu@accounts.mu.ac.in

To,

1	The Chairman, Board of Deans pvc@fort.mu.ac.in
2	**Faculty of Humanities,** **Offg. Dean** 1. Prof.Anil Singh Dranilsingh129@gmail.com **Offg. Associate Dean** 2. Prof.Manisha Karne mkarne@economics.mu.ac.in 3. Dr.Suchitra Naik Naiksuchitra27@gmail.com
	Faculty of Commerce & Management, **Offg. Dean,** 1 Prin.Ravindra Bambardekar principal@model-college.edu.in **Offg. Associate Dean** 2. Dr.Kavita Laghate kavitalaghate@jbims.mu.ac.in 3. Dr.Ravikant Balkrishna Sangurde Ravikant.s.@somaiya.edu 4. Prin.Kishori Bhagat kishoribhagat@rediffmail.com

	Faculty of Science & Technology **Offg. Dean** 1. Prof. Shivram Garje ssgarje@chem.mu.ac.in **Offg. Associate Dean** 2. Dr. Madhav R. Rajwade Madhavr64@gmail.com 3. Prin. Deven Shah sir.deven@gmail.com
	Faculty of Inter-Disciplinary Studies, **Offg. Dean** 1.Dr. Anil K. Singh aksingh@trcl.org.in **Offg. Associate Dean** 2.Prin.Chadrashekhar Ashok Chakradeo cachakradeo@gmail.com 3. Dr. Kunal Ingle drkunalingle@gmail.com
3	Chairman, Board of Studies,
4	The Director, Board of Examinations and Evaluation, dboee@exam.mu.ac.in
5	The Director, Board of Students Development, dsd@mu.ac.in DSW direcotr@dsw.mu.ac.in
6	The Director, Department of Information & Communication Technology, director.dict@mu.ac.in

AC – 28/03/2025
Item No. – 6.9 (N)

As Per NEP 2020

University of Mumbai



Syllabus for Major Vertical – 1 & 4		
Name of the Programme – B.Sc. (Information Technology)		
Faulty of Science and Technology		
Board of Studies in Information Technology		
U.G. Second Year Programme	**Exit Degree**	**U.G. Diploma in Information Technology**
Semester		**III & IV**
From the Academic Year		**2025-26**

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	**Title of program** **O:_____**	**B.Sc. (Information Technology)**
2	**Exit Degree**	**U.G. Diploma in Information Technology**
3	**Scheme of Examination** **R: _____**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	**Standards of Passing** **R: _____**	**40%**
5	**Credit Structure** **R. SU-510C** **R. SU-510D**	**Attached herewith**
6	**Semesters**	**Sem. III & IV**
7	**Program Academic Level**	**5.00**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2025-26**

Sd/-	**Sd/-**	**Sd/-**
Sign of the BOS Chairman **Dr. Srivaramangai R** **BOS in Information Technology**	**Sign of the Offg. Associate Dean** **Dr. Madhav R. Rajwade** **Faculty of Science & Technology**	**Sign of the Offg. Dean** **Prof. Shivram S. Garje** **Faculty of Science & Technology**

Under Graduate Diploma in Information Technology

Credit Structure (Sem. III & IV)

(B. Sc.)- Major & Minor

	R. SU-510C									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	III	8 Python Programming DATA STRUCTURES Operating System Major Practical 3	…	4	2	VSC:2, APPLIED MATHEMATICS	AEC:2	FP: 2 CC:2	22	
5.0	**R. SU-510D**									**UG Diploma 88**
	IV	8 Core Java Software Engineering Computer Networks Major Practical 4	…	4	2	SEC:2 Computer Graphics OR Mobile Programming	AEC:2	CEP: 2 CC:2	22	
	Cum Cr.	28		10	12	6+6	8+4+2	8+2+2	88	
Exit option; Award of UG Diploma in Major and Minor with 88 credits and an additional 4 credits core NSQF course/ Internship OR Continue with Major and Minor										

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Sem. - III

Vertical – 1
Major

Syllabus
B.Sc. (Information Technology)
(Sem.- III)

Title of Paper: Python Programming

Sr.No.	Heading	Particulars
1	**Description the course: Including but not limited to:**	**Introduction to Programming with Python** course is designed to help beginners learn Python, a versatile and beginner-friendly language known for its simplicity and readability. Python is an excellent choice for newcomers to programming due to its clear syntax and broad applications in fields like web development, data analysis, and artificial intelligence. In today's technology-driven world, programming skills are increasingly essential, and Python's popularity has surged due to its ease of use and extensive support community. Python is also a gateway language, allowing learners to transition easily into more advanced topics such as machine learning, data science, and web development. As an interpreted, high-level language, Python is particularly relevant across industries like technology, healthcare, finance, and academia, making Python proficiency a highly sought-after skill. The course focuses on core programming concepts like syntax, data structures, and control flow, ensuring that learners can write efficient and functional code. The course also encourages further learning, serving as a stepping stone for advanced Python courses or specialized areas like machine learning and web development. Python's beginner-friendly nature and expansive libraries make it an enjoyable language to learn, fostering both interest and engagement. By combining theory with hands-on projects, the course aims to spark curiosity and provide learners with tangible results from their efforts. As learners gain proficiency in Python, they will have the tools to tackle more complex programming challenges, making this course an invaluable starting point for anyone interested in programming or pursuing a career in tech. **Demand in the Industry:** Python's popularity in the industry is soaring. Professionals proficient in Python are in high demand across various sectors, including technology, finance, healthcare, and academia. Completion of this Course opens doors to entry-level positions in software development, quality assurance, data analysis, and scripting.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50

7	**Course Objectives (CO):** CO 1.Master the core features of Python, including its execution model and a wide range of data types. CO 2. Develop proficiency in control flow by working with conditional statements, loops and other control structures. CO 3. Work efficiently with arrays, strings, and complex data structures, leveraging Python's capabilities for data manipulation. CO 4. Apply functions, modules, and string operations to solve real-world programming problems with flexibility and ease. CO 5. Manage file operations, utilize regular expressions, and handle date and time functions for comprehensive Python programming tasks.	
8	**Course Outcomes (OC):** OC 1. Demonstrate mastery of Python features to tackle a wide range of programming challenges. OC 2. Utilize control flow statements to ensure accurate and logical program execution. OC 3. Efficiently manipulate arrays, strings, and data structures to enhance. data handling and problem-solving. OC 4. Design modular, efficient programs by leveraging functions, modules, and string operations. OC 5. Manage file operations, employ regular expressions, and manipulate date and time data to improve program functionality and performance.	
9	**Module 1:** **Basic Elements of Python Programming:** Features of Python, Execution of a Python Program, Python Interpreter, Comments, IDLE, Data types, Dictionary, Sets, Mapping, Basic Elements of Python, Variables, Input Function, Output Statements, Command Line Arguments. Operators, Precedence of Operators, Associativity of Operators **Control Statements:** The if statement, The if … else Statement, The if … elif … else Statement, Loop Statement- while loop, for loop, Infinite loop, Nested loop, The else suite, break statement, continue statement, pass statement, assert statement, return statement. **Arrays:** Creating Arrays, Indexing and Slicing of Arrays, Basic Array Operations, Arrays Processing, Mathematical Operations on Array, Aliasing Arrays, Slicing and Indexing in NumPy Arrays, Basic slicing, Advanced Indexing, Dimensions and Attributes of an Array **Functions:** Function definition and call, Returning Results, Returning Multiple Values from a Function, Built-in Functions, Difference between a Function and a Method, Pass Value by Object Reference, Parameters and Arguments, Recursive Functions, Anonymous or Lambda Functions. Modules in Python. Strings: Creating Strings, Functions of Strings, Working with Strings, Formatting Strings, Finding the Number of Characters and Words, Inserting Substrings into a String.	15 Hrs

	Module 2:	
	List: Exploring List, Tuples and Dictionaries: Lists, List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples, Functions in Tuple. **Working with Dictionaries:** Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries **Files in Python:** Opening and Closing a File, Working with Text Files, , Working with Binary Files, The 'with' statement, Pickle in Python, The seek() and tell() Methods, Random Accessing of Binary Files, Zipping and Unzipping Files, Working with Directories **Regular Expressions:** Introduction, Sequence Characters in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expression on Files, Retrieving Information from an HTML File **Date And Time in Python:** Time, Date, Date and Time Now, combining date and times, formatting date and time, Finding and comparing dates, Sorting dates, Knowing the Time taken by a Program, Working with Calendar Module	15 Hrs
10	**Books and References:** **Textbooks** 1. Learning Python, Fourth Edition by Mark Lutz Copyright © 2009 Mark Lutz. Published by O'Reilly Media, Inc. 2. Python Basics: A Practical Introduction to Python 3 Revised and Updated 4th Edition David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler **Reference Books** 1. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019 2. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018 3. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: DATA STRUCTURES

Sr.No.	Heading	Particulars
1	**Description the course:** **Including but Not limited to:**	Data Structures is a fundamental subject that focuses on the organization, storage, and manipulation of data. It provides the tools and techniques to efficiently manage and process data, forming the backbone of algorithms and software development.
2	**Vertical:**	Major
3	**Type:**	Theory
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** 1. To understand the fundamental concepts of data structures and their applications. 2. To analyze the efficiency of algorithms and operations on data structures. 3. To provide practical exposure to implementing data structures in programming. 4. To understand the properties and applications of arrays, linked lists, stacks, and queues. 5. To translate data structure concepts into working code using a programming language. 6. To apply data structures to solve real-world problems like searching and sorting. 7. To grasp the structure and traversal methods of binary trees and binary search trees.	
8	**Course Outcomes (OC):** **Students will be able to:** OC 1. Demonstrate knowledge of core data structures and their operations OC 2. Analyze the time and space complexity of algorithms and choose the most efficient solution for a given problem. OC 3. Translate algorithmic solutions into correctly functioning code using their chosen programming language. OC 4. Implement and traverse binary trees and binary search trees, demonstrating their understanding of these structures.	
9	**Module 1:**	
	1. Introduction Basic terminology: data, information, data structure, abstract data type (ADT) Classification of data structures: linear, non-linear Algorithm analysis: time complexity, Big O notation **2. Arrays and Linked Lists** Array representation and operations (traversal, insertion, deletion, searching) Linked lists: singly linked lists (representation, insertion, deletion, traversal) Comparison of arrays and linked lists, advantages and disadvantages. **3. Stacks and Queues** Stack ADT: push, pop, peek operations Array implementation of stacks Applications of stacks: expression evaluation (infix to postfix conversion) Queue ADT: enqueue, dequeue, peek operations Array implementation of queues Applications of queues: basic scheduling scenarios **4. Recursion** Concept of recursion, base case, recursive step Examples: factorial, Fibonacci sequence	15 Hrs

	Module 2:	
	1.Trees Binary trees: representation, traversal (inorder, preorder, post order) Binary search trees: insertion, deletion, search Applications of trees: basic hierarchical data representation **2.Hashing** Hash functions and hash tables Collision handling: separate chaining Applications of hashing: dictionaries **3. Sorting and Searching** Sorting algorithms: bubble sort, insertion sort, selection sort Searching algorithms: linear search, binary search	**15 Hrs.**
10	**Books and References:** 1. Data Structures and Algorithms made Easy: Data Structures and Algorithmic Puzzles, Narasimha Karumanchi ,5th Edition 2017 2. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar SPD,1st 2014 3. Problem Solving in Data Structures & Algorithms Using C by Hemant Jain ,1st Edition, BPB Publications, 2018 **4.** Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 4th Edition, MIT Press,2022	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 Hr. duration**
14	**Format of Question Paper:** (**Semester End Examination: 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: Operating System

Sr.No.	Heading	Particulars	
1	**Description the course :** **Including but Not limited to:**	Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file systems and protection) Introduce the issues to be considered in the design and development of operating system (memory, file and disk).	
2	**Vertical :**	**Major**	
3	**Type :**	Theory	
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)	
5	**Hours Allotted :**	30	
6	**Marks Allotted:**	30	
7	**Course Objectives(CO):** 1. Understand basic knowledge of computer operating system 2. structures and functioning. 3. Understand the process management mechanism 4. CO 3. Ability to apply CPU scheduling algorithms to manage tasks. 5. CO 4. Discuss methods of prevention and recovery from system deadlock 6. CO 5. Understand the implementation of file systems and directories		
8	**Course Outcomes (OC):** 1. Outline the basic concept of operating systems 2. Analyze the working of operating system 3. Examine the working of various scheduling approaches 4. Apply the concepts of synchronization and deadlock 5. Apply the file access mechanisms		
9	**Modules:-** **Module 1:**		
	Operating System Overview: Basics of operating systems: Generations, Types, Structure, Services, System Calls, System Boot, System Programs, Protection and Security. **Process Management**: Process Concepts, Process States, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues. **Process Synchronization:** Background, Critical-Section Problem, Peterson's Solution. Synchronization Hardware, Semaphores, Classic Problems of Synchronization.	**15 Hrs**	
	Module 2:		
	Memory Management: Main Memory, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing. **Deadlock:** System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock. **File System Interface:** File Concept, Access Methods, Directory Structure, and File System Structure.	**15 Hrs**	

10	**Books and Reference:** 1. Operating Systems – Internals and Design Principles William Stallings, Pearson 9th , 2009 2. Operating System Concepts, Abraham Silberschatz, Wiley, 8th Edition 3. Operating Systems, Godbole and Kahate, Godbole and Kahate, 3rd Edition.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: Major Practical 3

Sr.No.	Heading	Particulars
1	**Description the course: Including but not limited to:**	This course offers a comprehensive exploration of advanced Python programming concepts, designed to equip students with the tools to tackle real-world problems efficiently. It covers a range of topics, including text processing with regular expressions to identify patterns and extract meaningful data, as well as file handling techniques for both text and binary files. Students will also gain expertise in manipulating and comparing dates using Python's built-in date and time modules, along with performing calendar-based operations. The course emphasizes performance optimization by teaching students how to measure and improve program execution time. Additionally, students will learn how to extract structured data, such as hyperlinks from HTML files, and apply these techniques in practical scenarios. By the end of the course, students will be adept at solving complex problems, optimizing their Python solutions, and utilizing advanced programming concepts to handle diverse data processing tasks.
2	**Vertical:**	Major
3	**Type:**	Practical
4	**Credits:**	2 credits (30 Hours of Practical work in a semester)
5	**Hours Allotted:**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** 1. Understand fundamental programming concepts in Python, including input/output operations, conditional statements, and loops. 2. Understand and apply array operations, indexing, slicing, and mathematical functions using NumPy. 3. Develop problem-solving skills by using functions, recursive logic, lambda expressions, and modular programming. 4. Use data structures like lists and dictionaries and perform file operations. 5. Work with text processing, file handling, date manipulation, and performance analysis using advanced Python programming concepts 6. To provide hands-on experience in implementing fundamental data structures like arrays, linked lists, stacks, queues, trees, and graphs. 7. To develop skills in algorithm design and analysis for solving computational problems using data structures. 8. To enable students to choose appropriate data structures for different applications and justify their choices. 9. To enhance understanding of dynamic memory allocation and efficient data management techniques. 10. To equip students with the ability to debug and optimize code for data structure operations.	
8	**Course Outcomes (OC):** . OC 1. Apply Python programming concepts like input/output, conditional statements, and loops, to solve fundamental problems effectively. . OC 2. Demonstrate proficiency in performing basic operations, indexing, slicing, and analyzing attributes of arrays using NumPy. . OC 3. Apply functions, recursion, and lambda expressions to solve computational problems, and implement modular programming for reusable and efficient code design.	

	. OC 4. Implement lists and dictionaries, perform file operations, and use functions to solve real-world problems effectively. . OC 5. Process text, extract information, handle dates, and measure execution time for solving complex real-world problems. . OC 6. Ability to implement and manipulate basic and advanced data structures to solve real-world problems. . OC7 Proficiency in writing efficient algorithms using suitable data structures for operations like searching, sorting, and traversal. . OC8 Capability to analyze the time and space complexity of algorithms for various data structures. . OC9 Enhanced problem-solving skills by applying data structures in different domains such as databases, networks, and operating systems	
9	**Module 1** 1. Write programs for the following: a. Write a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old. b. Write a program to accept a number from the user and depending on whether the number is even or odd, print out an appropriate message to the user. c. Write a program to accept the SGPI from the user and print corresponding grade based on the following: d. SGPI Grade e. 9.00 – 10.00 O f. 8.00 – 8.99 A+ g. 7.00 – 7.99 A h. 6.00 – 6.99 B+ i. 5.50 – 5.99 B j. 5.00 – 5.49 C k. 4.00 – 4.99 P l. Below 4 F 2. Write programs for the following: a. d. Write a program to generate the Fibonacci series. b. e. Write a program to accept a number from the user display sum of its digits. 3. Write programs for the following: a. Write a program to perform basic operations, indexing and slicing on arrays. b. Write a program to implement mathematical functions on arrays. c. Write a program to perform array aliasing and copying. 4. Write programs for the following: a. Write a program to perform slicing, basic and advanced indexing on b. NumPy arrays. c. e.Write a program to analyze dimensions and attributes of arrays 5. Write programs for the following: a. Write a function to check the input value is Armstrong and also write the function for Palindrome. b. Write a recursive function to print the factorial for a given number. c. Write a lambda function that checks whether a given string starts with a specific character. 6. Write programs for the following: a. Write a program to compute number of characters and words in a string.	30 Hrs

	b. Create a file geometry.py to calculate base areas for shapes square and circle. In another file, write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas. 7. Write programs for the following: a. Write a program that takes two lists and returns True if they have at least one common member. b. Write a Python script to sort (ascending and descending) a dictionary by value. 8. Write programs for the following: a. Write a program to accept and pass radius to a function that returns area and circumference (using tuple). b. Write a program to perform basic file operations on text files and binary files. c. Write a Python program to read last n lines of a file. 9. Write programs for the following: a. a.Write a program to count the occurrences of a specific word in a file using regular expressions. b. b.Write a program to extract all hyperlinks (<a href="...">) from an HTML file. 10. Write programs for the following: a. Write a program that compares two dates (in DD/MM/YYYY format) and prints which one is earlier. b. Write a program to measure program execution time. c. Write a program using the calendar module to print the weekday of the first day of a given month and year.	
	Module 2	30 Hrs
	1. Array Operations: Write a program to implement basic array operations: a. Insert an element at a specific position in an array. b. Delete an element from a specific position in an array. c. Search for an element in an array (linear search). 2. Linked List Manipulation: Write a program to: a. Create a singly linked list. b. Insert a node at the beginning, end, and at a given position in a linked list. c. Delete a node from a given position in a linked list. 3. Stack Application: Write a program to: a. Implement a stack using an array. b. Convert an infix expression to postfix notation using a stack. 4. Queue Application: Write a program to: a. Implement a queue using an array. b. Simulate a simple queuing system (e.g., customer service queue). 5.Binary Search Tree: Write a program to: a. Create a binary search tree. b. Insert nodes into a binary search tree. c. Search for a node in a binary search tree. 6. Tree Traversal: Write a program to: a. Implement pre-order, b. in-order,	

	c. Post-order traversal of a binary tree. 7.Hash Table: Write a program to: a. Implement a hash table with separate chaining for collision handling. b. Store and retrieve data from the hash table. 8.Sorting Algorithms: Write programs to implement and compare the following sorting algorithms: a. Bubble sort b. Insertion sort c. Selection sort 9.Searching Algorithms: Write programs to implement and compare: a. Linear search b. Binary search (on a sorted array) 10.Combined Application a. Design a simple program that uses multiple data structures .	
10	Text Books: 1. Learning Python, Fourth Edition by Mark Lutz Copyright © 2009 Mark Lutz. Published by O'Reilly Media, Inc. 2. Python Basics: A Practical Introduction to Python 3 Revised and Updated 4th Edition David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler 3. Data Structures and Algorithms made Easy: Data Structures and Algorithmic Puzzles, Narasimha Karumanchi ,5th Edition 2017	
11	Reference Books: 1. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019 2. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018 3. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017 4. A Simplified Approach to Data Structures, Lalit Goyal, Vishal Goyal, Pawan Kumar SPD,1st 2014 5. Problem Solving in Data Structures & Algorithms Using C by Hemant Jain ,1st Edition, BPB Publications, 2018 6. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 4th Edition, MIT Press,2022	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Sem. - IV

Vertical – 1
Major

Title of Paper :Core Java

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	Core Java course focuses on teaching students how to design, develop, and maintain software applications using the Java programming language. The course covers fundamental to advanced concepts of Java, enabling students to understand object-oriented programming (OOP) principles, data structures, algorithms, and real-world application development.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted :**	30 Hr
6	**Marks Allotted:**	50
7	**Course Objectives(CO):** **CO 1:** Understand and Apply Object-Oriented Programming (OOP) Concepts. **CO 2:** Identify the key components of a class and object in Java, including attributes (fields), methods, and constructors. **CO 3:** Apply sound software engineering principles in Java by organizing code into classes and methods with proper access control identifiers **CO 4:** Use tools and techniques like unit testing, as well as IDE debugging tools to find and fix issues within Java programs. **CO 5.** Effectively use Java's collection framework (e.g., Lists, Sets, Maps) to manage and process groups of related objects. **CO 6.** Use OOP concepts in designing and building solutions to real-world problems, ensuring the application is modular, maintainable, and reusable.	
8	**Course Outcomes (OC):** OC1. Understand the basics of Java and its runtime environment. OC2. Be proficient in using Java's data types, control flow statements, and OOP principles such as classes, inheritance, and exception handling. OC3. Creating own classes and objects OC4. Develop mini projects using Class, Interface and exception handling	
9	**Modules:-** **Module 1:**	
	Introduction to Java Programming-History of Java and its Evolution,Features of Java (Platform Independence, Object-Oriented),Data Types and Variables,Operators Constants and Literals,Type Casting **Decision Making and Loops** :If-else Statements,Switch Statement, Loops (For, While, Do-While),Break and Continue Statements **Classes and Objects** :Array,ArraysString class and String methods, StringBuffer and StringBuilder, Object-Oriented Programming Concepts, Defining Classes and Creating Objects, Instance Variables and Methods, Constructors, this Keyword, super keyword, Types of Classes, Scope Rules, Access Modifier, constants, static members of a class, garbage collection.	**15 Hrs**

	Inheritance: Its types, Superclass and Subclass, Final classes and methods **Polymorphism**: Compile-time and Runtime Polymorphism	
	Module 2:	
	Interfaces: Defining and Implementing Interfaces,Abstract Classes and Methods,Multiple Interface Implementation **Packages**: Introduction to predefined packages, User Defined Packages, Access specifier, Java Built-in packages **Exception handling**- Try, Catch, and Finally Blocks, Throw and Throws Keywords **Introduction to Threads**: Creating and Running Threads,Thread Lifecycle	**15 Hrs**
10	**Books and References:** 1. Java: The Complete Reference Herbert Schildt MC-Graw HILL 12th EDITION 2022 2. Core Java, Volume I: Fundamentals Hortsman Pearson 9th 2013 3. Core Java, Volume II: Advanced FeaturesGary Cornell and Hortsman Pearson 8th 2008	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Title of Paper: Software Engineering

Sr.No	Heading	Particulars
1	**Description the course:** **Including but Not limited to:**	This course provides an in-depth understanding of Scrum, an Agile framework for developing, delivering, and sustaining complex products. Students will learn the principles and practices of Scrum, including roles (Scrum Master, Product Owner, Development Team), events (Sprint, Scrum Meetings), and artifacts (Product Backlog, Sprint Backlog, Increment). The course emphasizes hands-on exercises, real-world scenarios, and collaborative activities to master iterative development and enhance team productivity. By the end, learners will be equipped to implement Scrum in software engineering projects effectively and drive organizational agility.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted :**	30
6	**Marks Allotted:**	50
7	**Course Objectives (CO):** **CO1:** Understand the core principles of Agile and the Scrum framework. **CO2:** Explore the high-level Scrum process and its key components. **CO3:** Develop skills in managing the Product Backlog effectively. **CO4:** Learn techniques for Sprint planning, execution, and tracking. **CO5:** Gain insights into Scrum-based project, quality, and risk management. **CO6:** Master the art of writing clear and actionable user stories.	
8	**Course Outcomes (OC):** **OC1:** Demonstrate a comprehensive understanding of Agile concepts and Scrum practices. **OC2:** Apply Scrum processes to effectively manage software development life cycles. **OC3:** Create and prioritize user stories for efficient Product Backlog management. **OC4:** Utilize metrics to evaluate and enhance Sprint performance and team productivity. **OC5:** Implement strategies for cost, customer, and risk management in Scrum projects. **OC6:** Formulate effective Sprint retrospectives to drive continuous improvement.	
9	**Module 1:**	
	Software and Software Engineering, Process Models, Introduction to Agile Concepts, All about Scrum, Scrum Process: High-Level View. Product Backlog Management, Sprint Planning, Writing Effective User Stories, Sprint Execution and Tracking, Sprint Review, Sprint Retrospectives	15 Hrs
	Module 2:	
	Measurements and Metrics in Scrum, Software Development Life Cycle and Waterfall Model, Project Management in Scrum and Waterfall, Quality Management in Scrum, Customer Management in Scrum, Risk Management in Scrum, Cost Management in Scrum.	15 Hrs

<table>
<tr><td>10</td><td colspan="2">Books and References:
1. “Agile Scrum”, Rama Bedarkar, Wiley, 1st, 2020
2. "Mastering Professional Scrum: A Practitioner's Guide to Overcoming Challenges and Maximizing the Benefits of Agility" by Stephanie Ockerman and Simon Reindl, Addison-Wesley Professional, 1st edition (2019).
3. "Scrum: A Pocket Guide" by Gunther Verheyen, Van Haren Publishing, 2nd edition (2019).
4. "Software in 30 Days" by Ken Schwaber and Jeff Sutherland, Wiley, 1st edition (2012).
5. "Scrum Insights for Practitioners: The Scrum Guide Companion" by Hiren Doshi, PracticeAgile Solutions, 1st edition (2016).
6. "A Scrum Book: The Spirit of the Game" by Jeff Sutherland and James O. Coplien, Pragmatic Bookshelf, 1st edition (2019).
7. "The Scrum Fieldbook: A Master Class on Accelerating Performance, Getting Results, and Defining the Future" by J.J. Sutherland, Random House Business, 1st edition (2019).</td></tr>
<tr><td>12</td><td>Internal Continuous Assessment: 40%</td><td>Semester End Examination: 60%</td></tr>
<tr><td>13</td><td>Continuous Evaluation through:
Class test of 1 of 15 marks
Class test of 2 of 15 marks
Average of the two: 15 marks
Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td><td>Format of Question Paper: External Examination (30 Marks)– 1 hr duration</td></tr>
<tr><td>14</td><td colspan="2">Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)
Q1: Attempt any two (out of four) from Module 1 (15 marks)
Q2: Attempt any two (out of four) from Module 2 (15 marks)
Or
Q1: Attempt any three (out of five) from Module 1 (15 marks)
Q2: Attempt any three (out of five) from Module 2 (15 marks)</td></tr>
</table>

Title of Paper: Computer Networks

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	A course on **Computer Networks** typically focuses on the fundamental principles, technologies, and protocols that enable communication and data exchange between devices in various network environments.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted :**	30
6	**Marks Allotted:**	50
7	**Course Objectives(CO):** 1. To understand the basic concepts in OSI Model,distinguishing Factors in TCP/IP ,IP addressing Schemes 2. Understand How the communication happens across the network 3. Understanding of various Routing protocol and their implementation	
8	**Course Outcomes (OC):** CO 1.Understanding the Transport layer protocols and their utilities CO 2.Various application layer protocols and their implementation CO3:Mailing Services and web services implementation	
9	**Modules:-** **Module 1:**	
	1. Introduction: OSI Model, TCP/IP Protocol Suite, IPV 4 Addresses and Protocol and IPV6 Addresses and Protocol 2. Address Resolution Protocol (ARP), Internet Control 3. Message Protocol Version 4 (ICMPv4), Mobile IP, 4. Unicast Routing Protocols (RIP, OSPF and BGP)	**15 Hrs**
	Module 2:	
	8. User Datagram Protocol (UDP), Transmission Control Protocol (TCP) 9. Host Configuration: DHCP, Domain Name System (DNS) 10. Remote Login: TELNET and SSH, File Transfer: FTP and TFTP ; World Wide Web and HTTP, 11. Electronic Mail: SMTP, POP, IMAP and MIME	**15 Hrs**
10	**Books and References:** **TCP/IP Protocol Suite, Behrouz A. Forouzan, 4th Edition, Tata McGrawHill** (Chapter 2, 5, 7, 26, 27, Chapter 8, 9, 10, 11, Chapter 14,15, Chapter 16, 18, 19, Chapter 20, 21, 22, Chapter 23, 25)	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	Format of Question Paper: External Examination (30 Marks)– 1 hr duration

14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)

Title of Paper: Major Practical 4

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	Computer Networking Practical course focuses on providing hands-on experience with various networking concepts and techniques. Students typically practice configuring, troubleshooting, and testing network protocols and hardware in real-world scenarios. The practical component of this course emphasizes network setup, monitoring, and management skillsWireless Networks and Mobile Communications A Java Programming Practical course typically provides hands-on experience in writing, debugging, and executing Java programs. The goal is to help students become proficient in Java programming and apply theoretical concepts to solve real-world problems. The practical sessions in this course usually focus on programming skills and the application of Java principles in various scenario
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** 1. Understand core Java programming concepts, including data types, control structures, and object-oriented programming principles. 2. Develop the ability to implement inheritance, polymorphism, interfaces, and abstract classes effectively. 3. Gain hands-on experience with exception handling, multithreading, and dynamic initialization. 4. Learn to apply Java programming to solve real-world problems, such as matrix operations and finding areas/volumes. 5. Enhance debugging and problem-solving skills using Java's rich standard libraries and features. 6. Basic foundation of LAN 7. various command line utilities to be tested 8. Practical implementation of IP Subnetting 9. Testing of various Routing Protocols	
8	**Course Outcomes (OC):** CO 1. Write efficient Java programs to perform arithmetic operations, manage control flow, and manipulate strings. CO 2. Demonstrate knowledge of object-oriented concepts by implementing inheritance, polymorphism, and interfaces. CO 3. Apply exception handling mechanisms to create robust Java applications. CO 4.1Implement multithreading and explore dynamic initialization for advanced Java programming.	

	CO 5. Solve computational problems, such as matrix operations and factorial calculation, using packages and Java constructs. CO 6. Implementation of utility protocols CO 7 Understanding Basic Security features CO 8 Network Traffic and Packet Analysis CO 9 Basic Understanding of Wireless Network	
9	**Module 1** 1. Write a program a. in Java to demonstrate Boolean value. b. Print a string 10 times using a for loop. c. Write a program in Java to evaluate a+b*c%d. 2. Write a program a. in Java to find the biggest element among three numbers using if else. b. Write a program in Java to find the biggest element among three numbers using the ternary operator. c. Write a program in Java to check the grade of marks using a switch case. 3. Write a program a. in Java to demonstrate dynamic initialization. b. Write a program in Java to create a class and access all data members and methods using the object and compute the area and perimeter of a circle. c. Write a program in Java to access member variables using the constructor. 4. Write a program a. in Java to multiply two matrices. b. Write a program in Java to calculate the area of a rectangle using single inheritance. c. Write a program in Java to demonstrate multilevel inheritance. 5. Write a program a. in Java to demonstrate hierarchical inheritance. b. Write a program in Java to find the area and perimeter of a circle using an abstract class. c. Write a program in Java to show that a private member of a class cannot be inherited. 6.Write a program a. in Java to find the volume of a box using this keyword. b. Write a program in Java to find the average of three numbers using the method overloading c. Write a program in Java to find average of three numbers using method overriding. d. Create a class figure. Create two subclasses rectangle and triangle. Find the area of a rectangle and half the area of the rectangle using the reference of the figure. 7.Write a program a. Create an interface area. Find the area of a circle. b. Write a program in Java to find the sum and average of three numbers using the super keyword. 12. Write a program	**30 Hrs**

	a. in Java to find the volume of a box using constructor overloading. b. Write a program in Java to demonstrate exception handling in case of variable/constant divided by zero. 13. Write a program in Java a. to implement multiple inheritance using the interface. b. Write a program in Java to check if a given string is palindrome or not. 14. Write a program in Java a. for sorting a given list of strings in ascending order. b. Write a program in Java to find the factorial of a number using the package. 15. Write a program in a. Java to import the package. b. Write a program in Java to implement thread. c. Write program to implement Flow, Grid and Border Layout using swing. d. Write program to demonstrate following events Action Mouse Key	
	Module 2	**30 Hrs**
	1. Configuring LAN setup a. Planning and Setting IP networks b. Configuring subnet c. Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo 2. IPv4 Addressing and Subnetting a. Given an IP address and network mask, determine other information about the IP address such as: a. Network address • Network broadcast address • Total b. number of host bits • Number of hosts b. Given an IP address and network mask, determine other information about the IP address such as: c. The subnet address of this subnet • d. The broadcast address of this subnet • e. The range of host addresses for this subnet • f. The maximum number of subnets for this subnet mask • g. The number of hosts for each subnet • h. The number of subnet bits •The number of this subnet 3. Configure Static IP routing using . 4. Configure IP routing using RIP. 5. Configuring Simple and multi-area OSPF 6. Configuring BGP protocol (Multi-Autonomous) 7. Configuring server and client. a. Configure DHCP b. Configure DNS c. Configure HTTP d. Configure Telnet e. Configure FTP 8.Configure basic security features for networks 9.Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the	

	network analyzer is working 10.create a wireless network of multiple PCs using appropriate access point. 11.IPV6 Addressing Basics	
10 & 11	**Text Books& References Books :**	
	1. Java: The Complete Reference Herbert Schildt MC-Graw HILL 12th EDITION 2022 2. Core Java, Volume I: Fundamentals Hortsman Pearson 9th 2013 3. Core Java, Volume II: Advanced FeaturesGary Cornell and Hortsman Pearson 8th 2008 4. Cisco CCNA 200-301 Official Cert Guide	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Vertical - 4

VSC

Syllabus
B. Sc. (nformation Technology)
(Sem.- III)

Title of Paper APPLIED MATHEMATICS

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This course is designed for developing competency of the students in the applications of various mathematical concepts. It is equipped with Complex numbers, Laplace transform, Inverse Laplace transform, Differential equations of first order with first degree and higher degree. This course introduces basic concepts of Algebra and prepares students to study further courses in linear and abstract algebra.
2	**Vertical :**	Vocational Skill Course
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester, Total 30 hours)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1: Ability to interpret the mathematical results in physical or practical terms for complex numbers. CO2: Know and to understand various types of methods to solve Laplace transform. CO3: Apply the knowledge of Laplace Transforms to solve the problems. CO4: Know and to understand various types of methods to solve differential equation. CO5: Apply the knowledge of differential equations to solve the problems. CO6: Inculcate the habit of Mathematical Thinking through Indeterminate forms.	
8	**Course Outcomes (OC):** OC 1. Familiar with the various forms and operations of a complex number. OC 2: Find the Laplace transform of a function of using definition. OC 3: Find the Inverse Laplace transform of a function of using definition. OC 4: Solve Differential equations of first degree and first order. OC 5: Solve Differential equations of first degree and higher order.	
9	**Modules:-** **Module 1:**	
	1.1 **Complex Numbers**: Complex number, Equality of complex numbers, Graphical representation of complex number (Argand's Diagram), Polar form of complex numbers. Polar form of x+iy for different signs of x.y, Exponential form of complex numbers, Mathematical operation with complex numbers and their representation on Argand's Diagram, Circular functions of complex angles, Definition of hyperbolic function. Relations between circular and hyperbolic functions, Inverse hyperbolic functions. 1.2 **The Laplace Transform**: Introduction. Definition of the Laplace Transform, Table of Elementary Laplace Transforms. Theorems on Important Properties of	**15 Hrs**

	Laplace Transformation, First Shifting Theorem, Second Shifting Theorem, Convolution Theorem, Laplace Transform of Derivatives. 1.3 **Inverse Laplace Transform**: Shifting Theorem, Partial fractions Methods, Use of Convolution Theorem, Solution of Ordinary Linear Differential Equations with Constant Coefficients, Laplace Transformation of Special Function, Periodic Functions, Heaviside Unit Step Function, Dirac-delta Function (Unit Impulse Function).	
	Module 2:	
	2.1 **Equation of the first order and of the first degree**: Separation of variables, Equations homogeneous in x and y, Non-homogeneous linear equations, Exact differential Equation, Integrating Factor, Linear Equation and equation reducible to this form, Method of substitution. 2.2 **Differential equation of the first order of a degree higher than the first**: Introduction, Solvable for p (or the method of factors), Solve for y, Solve for x, Clairaut's form of the equation, Method of Substitution. 2.3 **Linear Differential Equations with Constant Coefficients**: Introduction, The Differential Operator, Linear Differential Equation f(D) y = 0, Different cases depending on the nature of the root of the equation f(D) = 0, Linear differential equation f(D) y = X, The complimentary Function, The inverse operator 1/f(D) and the symbolic expression for the particular integral,	**15 Hrs**
10	**Books and References:** 1. A text book of Applied Mathematies Vol I, P. N. Wartikar and J. N.Wartikar, Pune Vidyathi Griha,7* ,1995 2. A text book of Applied Mathematies Vol II , P. N. Wartikar and J. N. Wartikar, Pune Vidyathi Griha,7" .1995 3. Higher Engineering Mathematies, Dr. B. S.Grewal, Khanna Publications.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

SEC

Syllabus
B. Sc. (Information Technology)
(Sem.- IV)

Title of Paper: Computer Graphics

Sr.No	Heading	Particulars
1	**Description the course : Including but Not limited to:**	**Computer Graphics Practical** is a hands-on course designed to introduce students to the foundational principles of computer graphics, including 2D transformations, graphical modelling, and basic animations. The course focuses on applying mathematical concepts like translation, rotation, scaling, and shearing to create and manipulate graphical objects. Students will learn to use programming tools such as Python (Matplotlib, Pygame, OpenCV) or C/C++ libraries to implement these concepts. Through practical assignments, they will develop the skills to create simple 2D animations, simulate real-world objects, and design graphical scenes. The course bridges the gap between theoretical concepts and real-world applications, fostering creativity and problem-solving in visual computing.
2	**Vertical :**	Skill Enhancement Course
3	**Type :**	Practical
4	**Credits :**	2 credits (30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO1:** Introduce foundational concepts of 2D transformations, geometric modelling, and rendering techniques in computer graphics**.** **CO2:** Develop skills to apply 2D transformations (translation, rotation, scaling, shearing, reflection) and basic animations**.** **CO3:** Enable students to simulate real-world objects and create simple animations**.** **CO4:** Equip students to understand graphics pipelines, coordinate systems, and basic rendering principles**.** **CO5:** Foster creativity and logical thinking by implementing graphical scenes and animations.	
8	Course Outcomes (OC): OC1: Understand and apply 2D transformation matrices to graphical objects. OC2: Implement simple graphics primitives and manipulate them using transformations. OC3: Create basic 2D animations (e.g., bouncing ball, rotating shapes). OC4: Understand and utilize color models, coordinate systems, and graphical libraries. OC5: Develop basic graphical applications using lightweight tools and programming languages.	

9	**Module 1**	**30 Hrs**
	Module 1: Basic Setup and 2D Graphics Fundamentals 1. Installing Required Software • Objective: Install and configure Python, Matplotlib, OpenCV, or Pygame for computer graphics. • Task: Verify the installation and create a "Hello, Graphics!" window. 2. Drawing Basic Shapes • Objective: Draw lines, circles, rectangles, and polygons using graphical primitives. • Tool: Python with Matplotlib or OpenCV. 3. Line Drawing Algorithms • Objective: Implement the DDA (Digital Differential Analyzer) algorithm. • Tool: Python or C++. 4. Bresenham's Line Drawing Algorithm • Objective: Implement Bresenham's line drawing algorithm. • Tool: Python or C++. 5. Circle Drawing Algorithms • Objective: Implement the Midpoint Circle algorithm. • Tool: Python or C++. 6. Polygon Filling • Objective: Implement the boundary-fill and flood-fill algorithms. • Tool: Python or C++. 7. Translation Transformation • Objective: Shift a 2D object using translation matrices. • Tool: Python with Matplotlib. 8. Rotation Transformation • Objective: Rotate a 2D object about a fixed point or origin. • Tool: Python with Matplotlib. 9. Scaling Transformation • Objective: Scale a 2D object up or down using scaling matrices. • Tool: Python with Matplotlib. 10. Reflection Transformation • Objective: Reflect a 2D object across x-axis, y-axis, and diagonal. • Tool: Python with Matplotlib. 11. Shearing Transformation • Objective: Apply x-axis and y-axis shearing to a 2D object. • Tool: Python with Matplotlib. 12. Composite Transformations • Objective: Combine translation, rotation, and scaling on a 2D object. • Tool: Python with Matplotlib. 13. Clipping Algorithms • Objective: Implement the Cohen-Sutherland line clipping algorithm.	

	• Tool: Python or C++. 14. Window-to-Viewport Transformation • Objective: Map a 2D object from a window to a viewport. • Tool: Python or C++. 15. Basic Interactive Graphics • Objective: Create a simple interactive graphics program (e.g., moving a rectangle with arrow keys). • Tool: Python with Pygame.	
	Module 2	**30 Hrs**
	1. Simple Animation • Objective: Animate a moving ball across the screen. • Tool: Python with Pygame. 2. Bouncing Ball Animation • Objective: Create a bouncing ball with collision detection. • Tool: Python with Pygame. 3. Rotating Object Animation • Objective: Animate a rotating triangle or square. • Tool: Python with Pygame or Matplotlib. 4. Scaling Animation • Objective: Create an animation showing pulsating objects (grow/shrink). • Tool: Python with Matplotlib. 5. Multiple Object Animation • Objective: Animate multiple objects moving independently. • Tool: Python with Pygame. 6. Color Models • Objective: Experiment with RGB and HSI color models. • Tool: Python with OpenCV. 7. Bezier Curves • Objective: Draw and animate a Bezier curve. • Tool: Python with Matplotlib. 8. 2D Game Development Basics • Objective: Create a simple 2D game (e.g., a ball avoiding obstacles). • Tool: Python with Pygame. 9. Scene Creation • Objective: Design a basic 2D scene (e.g., a house, tree, and sun). • Tool: Python with Matplotlib. 10. Parallax Scrolling Animation • Objective: Implement parallax scrolling for a background in 2D graphics. • Tool: Python with Pygame. 11. Path Animation • Objective: Animate an object moving along a predefined path.	

	• Tool: Python with Matplotlib. 12. Collision Detection • Objective: Implement collision detection between 2D objects. • Tool: Python with Pygame. 13. Interactive Graphics with Mouse Input • Objective: Create an interactive program where shapes follow mouse clicks. • Tool: Python with Pygame. 14. Text Rendering • Objective: Render and animate text in a 2D graphical environment. • Tool: Python with Pygame. 15. Final Project • Objective: Combine multiple concepts to create a complete animated 2D scene. • Example: A car moving on a road with a rising sun and trees. Tool: Python with Matplotlib or Pygame	
10 & 11	**Reference and Text Books:** 1. Python Graphics: A Reference for Creating 2D and 3D Images, Bernard Korites, Apress, 2nd Edition 2023. 2. Computer Graphics from Scratch: A programmer's Introduction to 3D Rendering, Gabriel Gambetta, no starch press, 2021 3. 2D Computer Graphics: Modern C++ and Standard Library, Hakan Blomqvist, 2023	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Syllabus
B. Sc. (Information Technology)
(Sem.- IV)

Title of Paper Mobile Programming

Sr.No	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This course introduces the fundamentals of Flutter and Dart for building cross-platform mobile applications. Students will learn to create responsive user interfaces, manage app state, handle user inputs, and implement navigation and animations. The course also covers integrating APIs, working with databases, and deploying functional mobile apps for Android and iOS.
2	**Vertical :**	Skill Enhancement Course
3	**Type :**	Practical
4	**Credits :**	2 credits (30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO1:** Understand the fundamentals of Flutter and Dart programming for mobile app development. **CO2:** Learn how to set up the Flutter SDK and development environment. **CO3:** Develop skills to create basic Flutter applications using widgets like Text, Row, and Column. **CO4:** Explore the use of StatelessWidget and StatefulWidget for managing app states. **CO5:** Master the implementation of responsive UIs using MediaQuery and layouts. **CO6:** Gain knowledge of form creation, input handling, and validation in Flutter apps. **CO7:** Learn to navigate between screens and implement app navigation features like drawers. **CO8:** Understand how to use Flutter animations, including AnimatedContainer and FadeTransition. **CO9:** Explore database integration with APIs using packages like http and FutureBuilder. **CO10:** Build apps with themes, user interactions (e.g., taps and long presses), and custom styling.	

8	**Course Outcomes (OC):** **OC1:** Demonstrate the ability to configure Flutter and build a functional development environment. **OC2:** Create and run basic Flutter apps with appropriate UI components. **OC3:** Develop responsive and adaptive UIs for multiple screen sizes. **OC4:** Implement interactive features like counters, sliders, and switches in Flutter apps. **OC5:** Design and validate user input forms using TextFormField. **OC6:** Develop navigation flows between screens and integrate drawers for better usability. **OC7:** Create animations for smooth transitions and enhanced user experiences. **OC8:** Build applications that fetch and display data from public APIs asynchronously. **OC9:** Apply effective state management strategies to handle app states efficiently. **OC10:** Demonstrate the ability to debug, test, and optimize Flutter apps for deployment.	
9	**Module 1**	**30 Hrs**
	1. Install Flutter SDK on your computer and run the flutter doctor command to check your setup. 2. Create a "Hello, World!" Flutter application and run it on an emulator. 3. Modify the app's title and primary color in the MaterialApp widget. 4. Create a StatelessWidget that displays a greeting message. 5. Write a Dart program to calculate the sum of two numbers entered by the user. 6. Implement a Dart program that uses if-else statements to determine if a number is odd or even. 7. Demonstrate the use of a switch-case statement in Dart. 8. Write a program to print a multiplication table using a for loop. 9. Create a Flutter app with a Text widget that displays your name. 10. Build an app with a Column widget to arrange multiple Text widgets vertically. 11. Use a Row widget to arrange three buttons horizontally. 12. Create a Flutter app using Scaffold with an AppBar, Body, and a FloatingActionButton. 13. Create a simple counter app using StatefulWidget to increment and display a number. 14. Implement a TextField widget to accept user input and display it using a Text widget. 15. Design a Flutter app with a Container widget and customize its padding, margin, and color. Use the Stack widget to overlay a Text widget on an Image.	
	Module 2	**30 Hrs**
	1. Build a responsive UI using MediaQuery to adapt to different screen sizes. 2. Create a Flutter form with TextFormField widgets to accept a username and password. 3. Implement form validation to ensure the fields are not empty. 4. Add navigation between two screens in Flutter using the Navigator class.	

	5. Create a Drawer widget for app navigation with three menu options. 6. Display a list of items in a ListView widget. 7. Use the GridView widget to display a grid of images. 8. Add a GestureDetector to detect taps and display a message in the console. 9. Implement a LongPress event to change the color of a container. 10. Create a basic animation using the AnimatedContainer widget. 11. Implement a FadeTransition to animate the opacity of a widget. 12. Use a Slider widget to select a value between 0 and 100 and display the value. 13. Create a Switch widget to toggle between two themes (light and dark). 14. Use the http package to fetch and display data from a public API. Create a FutureBuilder widget to display data asynchronously.	
10 & 11	**Reference and Text Books:** 1. Mastering Flutter: A Beginner's Guide, by Sufyan bin Uzayr, CRC Press, 1st , 2023 2. Flutter for Beginners, by Alessandro Biessek, Packt Publishing, 1st edition (2019). 3. Flutter Cookbook, by Simone Alessandria, Packt Publishing, 2nd Edition, 2023 4. Beginning App Development with Flutter, by Rap Payne, Apress, 1st edition (2019). 5. Flutter Apprentice, by Michael Katz, Kevin David Moore, and Vincent Ngo, Kodeco, 1st edition (2021). 6. Flutter Complete Reference 2.0, by Alberto Miola, Independently published, 2nd edition (2023). 7. Flutter in Action, by Eric Windmill, Manning Publications, 1st edition (2020). 8. Programming Flutter, by Carmine Zaccagnino, O'Reilly Media, 1st edition (2020).	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks	

	Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

QUESTION PAPER PATTERN
(External and Internal)

Internal Continuous Assessment: 40%	**Semester End Examination: 60%**
Continuous Evaluation through: Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour) Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks) Or Q1: Attempt any three (out of five) from Module 1 (15 marks) Q2: Attempt any three (out of five) from Module 2 (15 marks)	

Practical Examination

Internal Continuous Assessment: 40%	**Semester End Examination: 60%**
Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 - 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	-	Ab (Absent)	0

Sd/-
Sign of the BOS Chairman
Dr. Srivaramangai R
BOS in Information Technology

Sd/-
Sign of the Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sd/-
Sign of the Offg. Dean
Prof. Shivram S. Garje
Faculty of Science & Technology

AC –24/11/2023
Item No. – 6.2 (N)

As Per NEP 2020

University of Mumbai



Title of the program

A- U.G. Certificate in Information Technology
B- U.G. Diploma in Information Technology
C- B.Sc. (Information Technology)
D- B.Sc. (Honours) in Information Technology
E- B.Sc. (Honours with Research) in Information Technology

Syllabus for Semester –

Sem I & II

Ref: GR dated 20th April, 2023 for Credit Structure of UG

(With effect from the academic year 2024-25 Progressively)

University of Mumbai



Syllabus for Approval

(As per NEP 2020)

Sr. No.	Heading	Particulars	
1	**Title of program** **O. _____A**	A	Title of the program U.G. Certificate in Information Technology
	O. _____B	B	U.G. Diploma in Information Technology
	O. _____C	C	B.Sc. (Information Technology)
	O. _____D	D	B.Sc. (Honours) in Information Technology
	O. _____E	E	B.Sc. (Honours with Research) in Information Technology
2	**Eligibility** **O. _____A**	A	10+2 (A learner must have completed HSC or equivalent with 45% of aggregate for open category and 40% of aggregate in case of reserved candidates in one attempt with Mathematics and/or Statistics as one of the subjects **(OR)** Passed Equivalent Academic Level 4.0 with CGPA equivalent to 45% for open category and 40% in case of reserved candidates with Mathematics and/or Statistics as one of the subjects
	O. _____B	B	Under Graduate Certificate in Information Technology Academic Level 4.5
	O. _____C	C	Under Graduate Diploma in Information Technology Academic Level 5.0
	O. _____D	D	Bachelors of Science in Information Technology with minimum CGPA of 7.5 Academic Level 5.5
	O. _____E	E	Bachelors of Science in Information Technology with minimum CGPA of 7.5 Academic Level 5.5
3	**Duration of program** **R. _____**	A	One Year
		B	Two Years
		C	Three years
		D	Four years

		E	Four years
4	**Intake Capacity** **R. _____**		
5	**Scheme of Examination** **R. _____**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination	
6	**Standards of Passing** **R. _____**	40% in each component	
7	**Sem. I & II Credit Structure** **R: _____A** **R: _____B** **Sem. III & IV Credit Structure** **R: _____C** **R: _____D** **Sem. V & VI Credit Structure** **R: _____E** **R: _____F**	Attached herewith	
8	**Semesters**	A	Sem I & II
		B	Sem I, II, III& IV
		C	Sem I, II, III, IV, V & VI
		D	Sem I, II, III, IV, V, VI, VII & VIII
		E	Sem I, II, III, IV, V, VI, VII & VIII
9	**Program Academic Level**	A	4.5
		B	5.0
		C	5.5
		D	6.0
		E	6.0
10	**Pattern**	Semester	
11	**Status**	New	
12	**To be implemented from Academic Year Progressively**	From Academic Year: 2023-24	

Sign of Chairperson
Dr. Mrs. R. Srivaramangai
Ad-hoc BoS (IT)

Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sign of Offg. Dean,
Prof. Shivram S. Garje
Faculty of Science & Technology

Preamble

1) Introduction

Information technology (IT) continues to be a dynamic and rapidly evolving field with high demand for skilled professionals. The demand for IT workers is driven by various factors, and the landscape may have evolved over a period of time. NEP envisages the multidisciplinary approach thus making IT much more applicable in all fields of life. This facilitates multi-institutional mobility of the students within India as well as abroad thus making the students attain different proficiency levels right from certificate to B.Sc Honours with Research. This new syllabus under NEP will thus enables the students for higher education, research and career in the field of IT

2) Aims and Objectives

The aims and objectives of a Bachelor of Science (B.Sc) program in Information Technology (IT) generally revolve around providing students with a comprehensive understanding of the principles, technologies, and applications within the field of information technology. The entire program collectively aim to produce graduates who are well-rounded IT professionals, capable of contributing to the design, development, and management of information technology systems in various industries. The specific details of the curriculum may vary among institutions offering B.Sc in Information Technology programs.

3) Learning Outcomes

The B. Sc. (Information Technology) Programme shall prepare and enable the graduates to:

- ✓ Demonstrate proficiency in programming languages, Data structures, Design and implement software solutions with their technical competence
- ✓ Analyze user requirements and design effective IT systems or applications.
- ✓ Apply system analysis and design methodologies to address complex business challenges.
- ✓ Acquire the skills of Database Management, Networking and Security, Web Technologies
- ✓ Plan, execute, monitor, and control IT projects.
- ✓ Analyze and solve complex IT problems using critical thinking skills.
- ✓ Apply concepts of artificial intelligence, machine learning, cloud computing, and IoT
- ✓ Effectively communicate technical information both orally and in writing.

4) Any other point (if any)

PROGRAMME SPECIFIC OUTCOMES (PSO)

On completing the B. Sc.(Information Technology) at the University of Mumbai, the graduates shall be able to

- Technical Proficiency:
  - Demonstrate a comprehensive understanding of fundamental concepts, principles, and technologies in information technology.
  - Apply programming and software development skills to design and implement IT solutions.
- System Thinking and Analysis:
  - Apply system analysis and design methodologies to analyze and address

complex problems.
    - Design and develop IT systems that meet user requirements and organizational needs.
- Database Management:
    - Design, implement, and manage relational databases to store and retrieve information effectively.
    - Demonstrate proficiency in using database management systems and querying languages.
- Networking and Security:
    - Understand and implement computer networks, protocols, and security measures.
    - Evaluate and implement security solutions to protect information systems.
- Web Technologies:
    - Develop web applications using a variety of technologies and programming languages.
    - Design and create user interfaces that adhere to web design principles.
- Project Management:
    - Apply project management principles to plan, execute, and deliver IT projects.
    - Demonstrate the ability to work effectively within project teams.
- Emerging Technologies:
    - Stay informed about and adapt to emerging technologies in the IT field.
    - Apply concepts of artificial intelligence, machine learning, cloud computing, and IoT to solve real-world problems.
- Critical Thinking and Problem-Solving:
    - Analyze and solve complex IT problems using critical thinking skills.
    - Apply problem-solving strategies to troubleshoot and resolve technical issues.
- Communication Skills:
    - Effectively communicate technical information to diverse audiences, both orally and in writing.
    - Collaborate with team members and stakeholders to achieve common goals.
- Ethics and Professionalism:
    - Demonstrate ethical behavior and professionalism in all aspects of the IT profession.
    - Adhere to ethical standards and legal considerations related to information technology.

5) Credit Structure of the Program (Sem I, II, III, IV, V and VI)
Under Graduate Certificate in Information Technology

(Credit Struture Sem I & II)

	R: _____A									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC, RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	I	**6**		**-**	**2+2**	**VSC:2, SEC:2**	**AEC:2, VEC:2, IKS:2**	**CC:2**	**22**	**UG Certificate 44**
		• Programming with C - 02 • Database Management Systems - 02 • Practical I - 02				**VSC :** Combinational and Sequential Design- 02 **SEC – 02** Office Tools for Data Management **OR** Fundamentals of Telecommunication Systems				
	R: _____B									
	II	**6**		**2**	**2+2**	**VSC:2, SEC:2**	**AEC:2,VEC:2**	**CC:2**	**22**	
		• OOPs with C++ - 02 • Web Designing - 02 • Practical II - 02				• VSC : Assembly Language Programming – 02 • SEC: 02 • Web Programming **OR** • PL/SQL				
	Cum Cr.	**12**	**-**	**2**	**8**	**8**	**10**	**4**	**44**	
Exit option: Award of UG Certificate in Major with 40-44 credits and an additional 4 credits core NSQF course/ Internship OR Continue with Majorand Minor										

Under Graduate Diploma in Information Technology

Credit Structure (Sem. III & IV)

	R: _____C									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC, RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	III	**8**		**4**	**2**	**VSC:2**	**AEC:2,**	**FP :2 CC:2**	**22**	
		• Python Programming -02 • Python Programming Practical-02 • Data Structures-02 • Data Structures Practical-02				VSC : Operating Systems-02				**UG Diploma 88**
	R: _____D									
	IV	**6**		**4**	**2**	**SEC:2**	**AEC: 2**	**CEP : 2 CC: 2**	**22**	
		• Core Java - 02 • Core Java Practical-02 • Software Engineering-02 • Software Engineering Practical-02				• Computer Graphics and Animation -02 **OR** • Mojo-02 **OR** • Mobile Programming-02				
	Cum Cr.	28		10	12	12	14	12	88	
Exit option; Award of UG Diploma in Major and Minor with 80-88 credits and an additional 4 credits core NSQF course/ Internship OR Continuewith Major and Minor										

B.Sc. (Information Technology)

Credit Structure (Sem. V & VI)

	R: _____E									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC, RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	V	**10**	**4**	**4**		**VSC: 2**		**FP/CEP:2**	**22**	
		• Advanced Web Programming-02 • Advanced Web Programming Practical-02 • Business Intelligence-02 • Business Intelligence Practical-02 • Software Project Management-02	• Linux Administration-02 • Linux Administration Practical-02 OR • EARN-02 • EARN Practical-02 OR • Enterprise Java-02 Enterprise Java Practical-02			• Advanced Mobile Programming-02		FP: Project Dissertation-02	22	**UG Degree 132**
	R: _____E									
	VI	**10**	**4**	**4**				**OJT:4**	**22**	
		• Security in Computing -02 • Security in Computing Practical-02 • AI and ML-02 • AI and ML Practical-02 • Software Quality Assurance-02	• Enterprise Networking-02 • Enterprise Networking Practical-02 OR • Principles of GIS-02 • Principles of GIS Practical-02					• OJT: Project Implementation-04		
	Cum Cr.	**48**	**8**	**18**	**12**	**14**	**14**	**18**	**132**	
Exit option: Award of UG Degree in Major with 132 credits OR Continue with Major and Minor										

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

SEMESTER I

Syllabus
B.Sc. (Information Technology)
(Sem.- I)

Major Courses

Name of the Course: Programming with C

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course allows the students to understand the fundamental concepts of programming which will allow them to program applications in C.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the concepts of computer programming. CO 2. To understand syntax and semantics of the C language CO 3. To understand loops and decision making in programming. CO 4. To understand the use of arrays, structures, union and pointers. CO 5. To understand functions for modular code and handle errors.	
8	**Course Outcomes (OC):** OC 1. Students can build flowcharts, pseudocode for C programs. OC 2. Students can use C language syntax and semantics in their programs. OC 3. Students can implement loops and decision making. OC 4. Students can use different types of data structures in their programs. OC 5. Students can write well-structured, readable, and maintainable C code and debug programs if there are any errors.	
9	**Modules:-** **Module 1:**	
	1. Introduction: Algorithms, History of C, Structure of C Program. Program Characteristics, Compiler, Linker and preprocessor, pseudo code statements and flowchart symbols, Desirable program characteristics. Program structure. Compilation and Execution of a Program, C Character Set, identifiers and keywords, data types and sizes, constants and its types, variables, Character and character strings, typedef, typecasting **2. Type of operators**: Arithmetic operators, relational and logical operators, Increment and Decrement operators, assignment operators, the conditional operator, Assignment operators and expression, Precedence and order of Evaluation Block Structure, Initialization, C Preprocessor	**15 Hrs**
	Module 2:	

	1. Control Flow: Statements and Blocks, If-Else, Else-If, Switch, Loops- While and For Loops Do-while, Break and Continue, Goto and Labels **2.** Basics of functions. User defined and Library functions **3.** Pointer and Addresses, Pointer and Function Arguments, Pointer and Arrays. **4.** User-defined data types- structure and union	**15 Hrs**
10	**Books and References:** 1. C Programming Language, Brian W. Kernighan, Dennis M. Ritchie , 2017 2. Let Us C, Yashvant Kanetkar, BPB Publications,2008. 3. Mastering in C, K. R. Venugopal and Sudeep R. Prasad, Tata McGraw-Hill Publications. 4. A Computer Science –Structure Programming Approaches using C, Behrouz Forouzan, Cengage Learning. 5.. Schaum's outlines Programming with C, Byron S. Gottfried, Tata McGraw- Hill Publications. 6. Basics of Computer Science, by Behrouz Forouzan, Cengage Learning. 7. Programming Techniques through C, by M. G. Venkateshmurthy, Pearson Publication.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Database Management System

Sr.No.	Heading	Particulars	
1	**Description the course : Including but Not limited to:**	The objective of the course is to present an introduction to fundamentals of database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively -information from a DBMS.	
2	**Vertical :**	Major	
3	**Type :**	Theory	
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory)	
5	**Hours Allotted :**	30 Hours	
6	**Marks Allotted:**	50 Marks	
7	**Course Objectives(CO):** CO 1. To make students aware fundamentals of database system. CO 2. To give idea how ERD components helpful in database design and implementation. CO 3. To experience the students working with database using MySQL. CO 4. To familiarize the student with normalization, database protection and different DDL, DML, DQL, DCL Statements CO 5. To make students aware about importance of protecting data from unauthorized users.		
8	**Course Outcomes (OC):** OC 1. Define and describe the fundamental elements of relational database management system. OC 2. To relate the basic concepts of relational data model, entity-relationship model, relational database OC 3. Design ER-models to represent simple database application scenarios. OC 4. Understand the normalization and its role in the database design process OC 5. Transform the ER-model to relational tables, populate relational database and formulate SQL OC 6. Understand basic database storage structures and access techniques: file and page organizations, indexing methods and hashing.		
9	**Modules:-** **Module 1:**		
	1. Introduction to Databases and transactions What is database system, purpose of database system, view of data, relational databases, database architecture, transaction management **2. Data Models** The importance of data models, Basic building blocks, Business rules, The evolution of data models, Degrees of data abstraction **3. Database Design, ER-Diagram** Database design and ER Model: overview, ER-Model, Constraints, ER-Diagrams, ERD Issues, Codd's rules, Relational Schemas **4. Relational database model** Logical view of data, keys, integrity rules		**15 Hrs**

	Module 2:	
	1. Database Design theory and normalization: Basics of functional dependencies and normalization for relational databases. Relational database design and further dependencies. **2. SQL, Indexing:** Introduction to SQL, Complex queries, triggers, views, joining database tables and schema modification. Query Processing and optimization. File structure, hashing and indexing **3. Transaction management and concurrency control and recovery:** Introduction to transaction processing concepts and theory. Concurrency control technique. Database recovery technique	. **15 Hrs**
10	**Text Books** 1. "Fundamentals of Database System", Elmasri Ramez, Navathe Shamkant, Pearson Education, Seventh edition, 2017 2. Database Management Systems", Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014 3. Database Systems: Design implementation and management by Carlos Coronel, Steven Morris, Peter Rob	
11	**Reference Books** 1. "Database System Concepts", Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill, 2017 2. "MySQL: The Complete Reference", Vikram Vaswani , McGraw Hill, 2017 3. "Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease", Ashwin Pajankar, BPB Publications, 2020	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Major Practical 1

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Programming with C -practical This course is stepping stone to learn other languages. This course provides students hands on experiences of coding exercises and projects. . Database Management System's practical approach is useful to gain the knowledge for software backend development. It benefits to user by providing data definition, data access, reduced data redundancy, data integrity, data sharing, data organizing, data consistency, data accuracy, and security
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (60 Hours of Practical work in a semester)
5	**Hours Allotted :**	30 Hours (C Programming Practical) + 30 Hours(DBMS - Practical)
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To provide exposure in developing algorithm, flowchart and to write efficient code. CO 2. To understand loops and decision making in programming. CO 3. To understand the arrays, structures, union. CO 4. To understand the use of function and pointers. CO 5. To Identify entities and its relationship with relational model structure. CO 6. To understand relational database using SQL and constraints implementation using create table queries. CO 7. To Understand DML operations and backing of database CO 8. To understand how to retrieve data from database and learn how to retrieve single value after performing calculations on group of values CO 9. To understand built-in functions to perform operations on data CO 10. To understand how to fetch data from two or more tables, which is joined to appear as single set of data CO 11. To understand nested and larger query as advanced fetching of data to understand concept of virtual table. CO 12. To understand how to control user access in a database.	

8	**Course Outcomes (OC):** OC 1. Students can demonstrate the concepts of datatypes, variables and operators in C. OC 2. Students can implement the concept of control statements and looping in C program. OC 3. Students can demonstrate the use of arrays, strings and structures in C OC 4. Students can implement modular C program using functions and pointers. OC 5. Students can demonstrate the use of arrays, strings and structures in C. OC 6. Students able to perform various operations such as insert, update delete and retrieve data from database using SQL queries. OC 7. Students able to perform alteration in tables and can restore and take backup of the database. OC 8. Students able to perform operations using simple SQL Queries to fetch data and learns various aggregate functions to get single value. OC 9. Students able to perform SQL Queries using JOIN keyword for joining two or more tables. OC 10. Students able to perform nested queries using in, exists operators. OC 11. Students able to create new table by joining one or more tables and learn how to hide attribute from end user. OC 12. Students able to restrict the user from accessing data in database. OC 13. Students should be able to create, manipulate the database management system to evaluate the business information problem.

	9	**Module 1:- Programming with C** **1. Practical 1**:- a. To calculate simple interest taking principal, rate of interest and number of years as input from user. Write algorithm & draw flowchart for the same. b. Write a program to find greatest of three numbers using conditional operator. Write algorithm & draw flowchart for the same. c. Write a program to check if the year entered is leap year or not. Write algorithm & draw flowchart for the same. **2. Practical 2:-** a. Write a program to calculate roots of a quadratic equation. b. Write a menu driven program using switch case to perform add / subtract / multiply / divide based on the users choice. c. Write a program to print the pattern of asterisks. **3. Practical 3** a. Write a program using while loop to reverse the digits of a number. b. Write a program to calculate the factorial of a given number. c. Write a program to print the Fibonacci series. **4. Practical 4** a. Write a program to print area of square using function. b. Write a program using recursive function. c. Write a program to square root, abs() value using function. d. Write a program using goto statement . **5. Practical 5** a. Write a program to print rollno and names of 10 students using array. b. Write a program to sort the elements of array in ascending or descending order **6. Practical 6** a. Write a program to extract the portion of a character string and print the extracted part. b. Write a program to find the given string is palindrome or not. c. Write a program to using strlen(), strcmp() function . **7. Practical 7** Write a program to swap two numbers using a function. Pass the values to be swapped to this function using call-by-value method and call-by-reference method. **8. Practical 8** a. Write a program to read a matrix of size m*n. b. Write a program to multiply two matrices using a function. **9. Practical 9** Write a program to print the structure using Title Author Subject Book ID Print the details of two students. **10. Practical 10** Create a mini project on “Bank management system”. The program should be menu driven.	**30 Hrs**

		Module 2 1. Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) 2. Perform the following: • Viewing all databases • Creating a Database • Viewing all Tables in a Database • Creating Tables (With and Without Constraints) • Inserting/Updating/Deleting Records in a Table 3. Perform the following: • Altering a Table • Dropping/Truncating/Renaming Tables • Backing up / Restoring a Database 4. Perform the following: • Simple Queries • Simple Queries with Aggregate functions 5. Queries involving • Date Functions • String Functions • Math Functions 6. Join Queries • Inner Join • Outer Join 7. Subqueries • With IN clause • With EXISTS clause 8. Converting ER Model to Relational Model and apply Normalization on database. (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys and normalization up to 3rd Normal Form). 9. Views • Creating Views (with and without check option) • Dropping views • Selecting from a view 10. DCL statements • Granting and revoking permissions • Saving (Commit) and Undoing (rollback)	**30 Hrs**
10	**Text Books:** 1.“Fundamentals of Database System”, Elmasri Ramez, Navathe Shamkant, Pearson Education, Seventh edition, 2017 . 2.Database Management Systems”, Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014		
11	**Reference Books:** 1. MASTERING C, K. R. Venugopal and Sudeep R. Prasad, Tata McGraw-Hill Publications. 2. “A Computer Science –Structure Programming Approaches using C”, Behrouz		

	Forouzan, Cengage Learning. 3. Schaum's outlines "Programming with C", Byron S. Gottfried, Tata McGraw-Hill Publications. 4. "Basics of Computer Science", Behrouz Forouzan , Cengage Learning. 5. "Programming Techniques through C", M. G. Venkateshmurthy, Pearson Publication. 6. "Programming in ANSI C", E. Balaguruswamy, Tata McGraw-Hill Education. 7. "MySQL: The Complete Reference", Vikram Vaswani , McGraw Hill, 2017. 8. "Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease", Ashwin Pajankar, BPB Publications, 2020.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Vocational Skill Course (VSC)

Name of the course: Combinational and Sequential Design

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Combinational and Sequential Design is a course that focuses on digital electronics and the design of circuits that combine multiple digital components. The course covers the theoretical and practical aspects of both combinational and sequential circuit design, as well as their applications. Digital circuits are used in many electronic devices, including computers, smartphones, and communication systems. The design of these circuits is critical to the performance and functionality of these devices. Understanding the basics of combinational and sequential design is essential for anyone interested in pursuing a career in the field of digital electronics. The course will cover the various techniques and tools used in digital circuit design, including Boolean algebra and K-map simplification. The course is highly relevant in today's technological landscape, as all modern electronics devices are based on digital circuits. The skills learned in the course are highly useful in various fields, such as computer and electronics engineering, telecommunications, and robotics. The application of combinational and sequential design is quite broad, and the skills acquired from the course can be applied in various areas. Students will be able to design digital circuits, troubleshoot and repair digital circuits, and optimize circuit performance. The course is highly interesting and engaging, providing students with the opportunity to explore and analyze complex digital circuitry. It is also connected to other courses such as Digital Logic Design, Computer Organization, and Microcontrollers. The demand for professionals with digital circuit design skills is high in various industries such as electronics, semiconductors, telecommunications, and computing. There is an increasing demand for professionals with these skills,

		and job prospects are promising for those with a solid background in digital circuit design. In summary, Combinational and Sequential Design is a course that offers students a comprehensive understanding of digital circuits' design principles and techniques. The knowledge and skills gained from this course are highly useful and applicable in various industries, with promising career prospects.
2	**Vertical :**	Vocational Skill Course(VSC)
3	**Type :**	Practical
4	**Credits :**	2 credits (60 hours in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To provide students with a comprehensive understanding of combinational and sequential circuit design principles and techniques. CO 2. To enable students to apply Boolean algebra, K-map simplification, and other design techniques to create optimized digital circuits. CO 3. To equip students with the necessary tools and skills to implement arithmetic circuits, data path circuits, and memory circuits. CO 4. To enable students to analyze and troubleshoot digital circuits to ensure optimal performance. CO 5. To provide students with hands-on practical experience in designing and implementing digital circuits using simulation software and real-world hardware.	
8	**Course Outcomes (OC):** OC 1. Students can explain the differences between combinational and sequential circuits, and identify their different applications. OC 2. Students can define the concept of Boolean algebra and its importance in digital circuit design. OC 3. Students can explain and apply the principles of K-map simplification and other design techniques. OC 4. Students can design and construct combinational circuits using Boolean algebra and K-maps. OC 5. Students can design and implement arithmetic circuits such as adders, subtractors, and multipliers. OC 6. Students can design and implement data path circuits such as registers, multiplexers, and decoders. OC 7. Students can implement digital circuits using breadboards, logic probes, and oscilloscopes. OC 8. Students can troubleshoot and verify the correctness of digital circuits using real-world hardware and measure their performance using various metrics.	
9	**Modules:-** **Module 1:**	

		1. Study of Logic gates and their ICs and universal gates: a. Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates b. Study of IC 7400, 7402, 7404, 7408, 7432, 7486, 74266 c. Implement AND, OR, NOT, XOR, XNOR using NAND gates. d. Implement AND, OR, NOT, XOR, XNOR using NOR gates. **2. Implement the given Boolean expressions using minimum number of gates.** a. Verifying De Morgan's laws. b. Implement other given expressions using minimum number of gates. c. Implement other given expressions using minimum number of ICs. **3. Implement combinational circuits.** a. Design and implement combinational circuit based on the problem given and minimizing using K-maps. (Various Equations, SOP, POS forms can be given) **4. Implement code converters.** a. Design and implement Binary – to – Gray code converter. b. Design and implement Gray – to – Binary code converter. c. Design and implement Binary – to – BCD code converter. d. Design and implement Binary – to – XS-3 code converter. **5. Implement Adder and Subtractor Arithmetic circuits.** a. Design and implement Half adder and Full adder. b. Design and implement BCD adder. c. Design and implement XS – 3 adder. d. Design and implement binary subtractor. e. Design and implement BCD subtractor. b. Design and implement XS – 3 subtractor.	**30 Hrs**
	Module 2:		
		6. Implement Arithmetic circuits. a. Design and implement a 2-bit by 2-bitultiplier. b. Design and implement a 2-bit comparator. **7. Implement Encode and Decoder and Multiplexer and Demultiplexers.** a. Design and implement 8:3 encoder. b. Design and implement 3:8 decoder. c. Design and implement 4:1 multiplexer. Study of IC 74153, 74157 d. Design and implement 1:4 demultiplexer. Study of IC 74139 e. Implement the given expression using IC 74151 8:1 multiplexer. f. Implement the given expression using IC 74138 3:8 decoder. **8. Study of flip-flops and counters.** a. Study of flip-flops and counters. b. Study of IC 7473. c. Study of IC 7474. d. Study of IC 7476. e. Conversion of Flip-flops.	**30 Hrs**

	f. Design of 3-bit synchronous counter using 7473 and required gates. g. Design of 3-bit ripple counter using IC 7473. **9. Study of counter ICs and designing Mod-N counters.** a. Study of IC 7490, 7492, 7493 and designing mod-n counters using these. b. Designing mod-n counters using IC 7473 and 7400 (NAND gates) **10. Design of shift registers and shift register counters.** a. Design serial – in serial – out, serial – in parallel – out, parallel – in serial – out, parallel – in parallel – out and bidirectional shift registers using IC 7474. b. Study of ID 7495. c. Implementation of digits using seven segment displays.	
10	**Text Books** 1. Digital Electronics and Logic Design, N. G. Palan, Technova	
11	**Reference Books** 1. Digital Principles and Applications, Malvino and Leach, Tata McGrawHill 2. Modern Digital Electronics, R. P. Jain, Tata McGrawHill 3. Digital Design, M. Morris R. Mano, Michael D. Ciletti, Pearson Education, 2012	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Skill Enhancement Course (SEC)

Name of the Course: Office Tools for Data Management

Sr.No.	Heading	Particulars
1	**Description the course:**	• **Introduction:** The MS Access course offers a comprehensive understanding of Microsoft's relational database management system, making it a valuable skill in today's data-driven environment. This course is designed to empower individuals with the tools needed to efficiently organize, manage, and analyse data. • **Relevance and Usefulness:** It provides practical insights into leveraging a relational database system for enhanced efficiency and organization. The MS Access course is useful for individuals seeking to enhance their data management skills. • **Applications:** With applications in various sectors, from business to research and project management, MS Access is versatile. It facilitates the creation of databases for tasks ranging from contact management to complex systems for inventory and financial analysis. • **Interest and Connection with Other Courses:** Its practical applications and user-friendly interface make it attractive to individuals looking to boost their data management skills. The MS Access course establishes a practical link with other data-related courses, offering foundational knowledge in database management. It complements courses in data analysis, business intelligence, and information systems. • **Demand in the Industry:** As businesses increasingly rely on data for decision-making, there is a growing demand for professionals skilled in database management. Proficiency in MS Access is particularly sought after in roles involving data organization, analysis, and reporting. • **Job Prospects:** Professionals completing the MS Access course are well-positioned for roles requiring efficient data management and analysis. Job prospects include positions in database administration, data analysis, and business intelligence, where MS Access proficiency is a valuable asset.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Practical
4	**Credits :**	2 credits
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):**	

	CO 1. Participants will grasp essential database concepts, including tables, relationships, and normalization principles. CO 2. Participants will design and construct well-organized databases in MS Access, showcasing proficiency in table design and data validation. CO 3. Participants will master the creation of complex queries in MS Access, enabling them to extract specific information efficiently. CO 4. Participants will develop expertise in crafting user-friendly forms and interfaces in MS Access, optimizing data entry processes. CO 5. Participants will generate comprehensive reports in MS Access, demonstrating skills in grouping, sorting, and presenting data for meaningful analysis.	
8	**Course Outcomes (OC):** OC 1. Participants can explain normalization importance, identify table relationships, and justify database design choices. OC 2. Participants create well-structured MS Access databases with proper relationships, data types, and normalization. OC 3. Participants execute advanced queries in MS Access, retrieving specific information based on diverse criteria. OC 4. Participants design intuitive MS Access forms, incorporating controls for an efficient and user-friendly data entry experience. OC 5. Participants produce insightful MS Access reports, organizing and presenting data effectively for analysis.	
9	**Modules:- All Practicals are based on MS Access** **Module 1:**	
	Practical 1: A. Getting familiar with MS Access Ribbon options. B. With the help of access wizard Create Database. Add 2 Tables. In each table add 5 columns of different data types. Add 10-10 entries in each table. Add necessary integrity constraints. C. Use the Table Wizard to create a table. Add and delete fields in an existing table. Establish an input mask and validation rule for fields within a table. Switch between the Design and Datasheet views of a table. **Practical 2:** A. Create and use an Input Mask to enter the data in sample table. B. Adding records in table by using Datasheet View, using a Form and using SQL. C. Create the Employee Database with necessary table and data and then implement the following transitions: • Delete the record for Kelly Marder. • Change Pamela Milgrom's salary to $59,500. • Use the Replace command to change all occurrences of "Manager" to "Supervisor". **Practical 3:** A. Create the Bookstore database with necessary table and data and modify the database to accommodate the following:	**30 Hrs**

		i. Add the book Exploring Microsoft Office 2000 Vol II (ISBN: 013-011100-7) by Grauer/Barber, published in 1999 by Prentice Hall, selling for $45.00. ii. Change the price of Memory Management for All of Us to $29.95. iii. Delete The Presentation Design Book. B. Create a table employ with (idno, name, job, age, salary). Insert 10 records. Create a query to display the information of all managers. Create a query to display the names of employs who"s salary is >15000. C. Use the Form Wizard to create a form, Move and size controls within a form. Use the completed form to enter data into the associated table. **Practical 4:** A. Add fields to an existing table. Use the Lookup Wizard to create a combo box. Add controls to an existing form to demonstrate inheritance. Add command buttons to a form. B. Generate and use various the queries using Query Wizards. C. Generate and use various the Query with User Input. D. Demonstrate use of Expression Builder. **Practical 5:** A. Use the report wizard to create a new report. Modify an existing report by adding, deleting, and/or modifying its controls. B. Create a query containing a calculated control. Then, create report based on that query. Use the Sorting and Grouping command to add a group header and group footer to a report. C. Use action queries to modify a database. Create a crosstab query to display summarized values from a table.	
		Module 2:	
		Practical 6: A. Create and Open a database with multiple tables; Identify the one-to-many relationships within the database and to produce reports based on those relationships. B. Create and Open a database with multiple tables; Identify the one-to-one relationships within the database and to produce reports based on those relationships. C. Create and Open a database with multiple tables; Identify the Many-to-Many relationships within the database and to produce reports based on those relationships. **Practical 7:** A. Demonstrate use of look up tables. B. Use the Report Wizard to create a report having the following requirements: i. Select the LastName field from the Author table. ii. Select the Title and Price fields from the Book table. iii. Select the PubName field from the Publisher table.	**30 Hrs**

	iv. View the data by Publisher. v. Add a grouping level using LastName. vi. Sort the report by the Title field in ascending order. vii. Choose Stepped layout and Portrait orientation. viii. Type Book List as the report's title. C. Define the relationship between two tables and add a subform to a form. **Practical 8:** A. Import an Access table from an Excel workbook. Create a one-to-many relationship between tables in a database. Create a multiple-table query. B. Import external data from the Excel spreadsheet file Bookstore. i. Make sure Import the source data into a new table in the current database is selected. ii. Select the Author worksheet. iii. Make sure that First Row Contains Column Headings is selected. iv. For the AuthorID field, set the Data Type option to Long Integer and set the Indexed option to Yes (No Duplicates). v. Select Choose my own primary key and make sure the AuthorID field is selected. vi. Save the table with the name Author. C. Export data from access to various formats. **Practical 9:** A. Relationships: Create and Use Author and Book Table. i. Create a relationship between the AuthorID field in the Author table and the AuthorCode field in the Book table. Put a checkmark in the box labeled Enforce Referential Integrity. ii. Create a relationship between the PubID field in the Publisher table and the PubID field in the Book table. Put a checkmark in the box labeled Enforce Referential Integrity. B. Create a switchboard; Use the Link Tables command to associate tables in one database with the objects in a different database. C. Create an AutoExec and a Close Database macro and demonstrate the use. **Practical 10:** A. Create the College Library database find out the following: - i. Total no. of copies of books subject wise. ii. A report displays all books group by Publisher. iii. A report displays all books group by Book Title. iv. A report displays all books group by Book Edition B. Demonstrate the use of Database Splitter Wizard by splitting database. C. Make Access database as an executable-only	
10	**Online reference/Text Books** 1. https://www.quackit.com/microsoft_access/tutorial/	

	2. https://www.tutorialspoint.com/ms_access/index.htm 3. Access 2016 in easy steps, by Mike McGrath, In Easy Steps, 1st Edition, 2017 4. Relational Databases and Microsoft Access, by Ron McFadyen, 1st Edition	
11	**Reference Books** 1. MICROSOFT ACCESS 2019 by David Murray, Kendall Hunt Publishing, 1st Edition, 2020. 2. Step by Step Microsoft Access 2013, by Joyce Cox and Joan Lambert, 1st Edition, Microsoft Press, 2013 3. Access 2019 Bible, by Michael Alexander, Richard Kusleika, Wiley, 1st Edition, 2018 4. Access 2019 For Dummies, by Laurie A. Ulrich, Ken Cook, Wiley, 1st Edition, 2018	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Name of the Course: Fundamentals of Telecommunication Systems

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	The course on Fundamentals of Telecommunication Systems aims to provide an in-depth understanding of the basic concepts and theories of signals and systems, as well as their applications in the field of telecommunication engineering. The course also focuses on the latest trends in 5G technology, providing students with insights into the driver, pillars, and challenges of 5G networks. Relevance and Usefulness: The course is highly relevant to students pursuing degrees in electronics and communication engineering, as well as those interested in telecommunications engineering. By focusing on key concepts and terminologies, such as sets, mappings, functions, and systems operators, the course provides a foundation for understanding both the theoretical and

		practical aspects of signals and systems. Additionally, the course helps students understand the role of 5G technology in enabling advanced wireless communication and the internet of things (IoT), which can be useful for developing innovative applications and services. Application and Interest: By completing the course, students will be equipped to apply their knowledge and skills in a range of industries and sectors, including telecommunication, internet of things, and wireless communication. The course is also highly engaging, as it covers several fascinating topics, including wireless communication, 5G technology, and IoT, among others. Connections with Other Courses: The course has links with other courses in electronics and communication engineering, including digital signal processing, telecommunication theory and practice, mobile communication, Information Technology and internet of things. Demand in the Industry and Job Prospects: Graduates with a background in signals and systems and 5G technology are in high demand in the telecommunication industry, as there is an increasing need for professionals who can design, implement, and oversee advanced communication networks. Specializations in 5G technology and signals and systems can open up a range of job prospects, including roles such as telecommunications engineer, network architect, systems engineer, and wireless communication developer, among others. In conclusion, the course in signals and systems and 5G technology is highly relevant and useful for students pursuing degrees in electronics and communication engineering and Information Technology. The course is engaging and provides a solid foundation in key concepts and technologies, enabling students to pursue a range of job prospects within the telecommunication industry.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Theory
4	**Credits :**	2 credits (30 hours in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. Identify the fundamental concepts and terminologies of signals and systems theory through an introduction to sets, mappings, functions, and systems operators.	

	CO 2. Demonstrate knowledge of the properties of continuous-time signals and systems, transformations of the independent variable, properties of functions, and representation of arbitrary functions. CO 3. Demonstrate knowledge of the properties of discrete-time signals and systems, transformations of the independent variable, properties of sequences, and representation of arbitrary sequences. CO 4. Analyze the drivers for 5G technology, identify the 10 pillars of 5G, and describe the evolution of wireless communication from LTE technology to beyond 4G. CO 5. Discuss the 5G internet of things (IoT), explain networking reconfiguration and virtualization support, and identify the mobility and quality of service control in 5G networks. CO 6. Evaluate the challenges of small cells in 5G mobile networks and identify the capacity limits and achievable gains with densification.	
8	**Course Outcomes (OC):** OC 1. Identify the fundamental concepts and terminologies of signals and systems theory through an introduction to sets, mappings, functions, and systems operators. OC 2. Demonstrate knowledge of the properties of continuous-time signals and systems, transformations of the independent variable, properties of functions, and representation of arbitrary functions. OC 3. Demonstrate knowledge of the properties of discrete-time signals and systems, transformations of the independent variable, properties of sequences, and representation of arbitrary sequences. OC 4. Analyze the drivers for 5G technology, identify the 10 pillars of 5G, and describe the evolution of wireless communication from LTE technology to beyond 4G. OC 5. Discuss the 5G internet of things (IoT), explain networking reconfiguration and virtualization support, and identify the mobility and quality of service control in 5G networks. OC 6. Evaluate the challenges of small cells in 5G mobile networks and identify the capacity limits and achievable gains with densification.	
9	**Modules:-** **Module 1: Signals and Systems:** 1. **Signals and Systems:** Introduction, Signals, Systems, Why Signals and Systems? Preliminaries, Overviews, Sets, Mappings, Functions, Sequences, Abuse of notations, System operators, Basic Signal Properties. 2. **Continuous-Time Signals and Systems:** Overview, Transformations of the Independent Variable, Transformations and the Dependent Variable, Properties of functions, Elementary functions, Representation of Arbitrary Functions using elementary functions, Continuous -time systems, Properties of systems, 3. **Discrete-Time Signals and Systems:** Overview, Transformations of the independent variable, Properties of Sequences, Elementary Sequences, Representing Arbitrary Sequences Using Elementary Sequences, Discrete-Time Systems, Properties of Systems	**15 Hrs**
	Module 2: Fundamentals of 5G Networks	
	4. **Drivers for 5G:** Introduction, Historical trend of Wireless Communication, Evolution of LTE technology to beyond 4G, 5G	**15 Hrs**

	Roadmap, 10 pillars of 5G, 5G in Europe, 5G in Asia, 5G in Asia, 5G Architecture 5. **The 5G Internet:** Introduction, Internet of Things and Context-Awareness, Networking Reconfiguration and Virtualisation Support, Mobility, Quality of Service Control, Emerging Approach for Resource Over-Provisioning 6. **Small Cells for 5G Mobile Networks:** Introduction, What are small cells? Capacity Limits and Achievable Gains with Densification, Mobile Data Demand, Demand vs Capacity, Small-Cell Challenges, Conclusions and future directions	
10	**Text Books:** 1. Signals and Systems, Michael Adams, University of Victoria, 3rd Edition, 2012 2. Fundamentals of 5G Mobile Networks, Edited by Jonathan Rodriguez, Wiley Publications, 2015	
11	**Reference Books** 1. Signals and Systems, Michael Adams, University of Victoria, 3rd Edition, 2012 2. Fundamentals of 5G Mobile Networks, Edited by Jonathan Rodriguez, Wiley Publications, 2015	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

SEMESTER II

Syllabus
B.Sc. (Information Technology)
(Sem.- II)

Major Courses

Name of the Course: Object Oriented Programming using C++

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course provides students knowledge and skills to understand and implement the object oriented skills. It will help them to implement OOP solutions to real-world problems.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To explain the difference between object oriented programming and procedural programming. CO 2. To understand OOP principles to create modular, reusable, and maintainable code. CO 3. To understand the concept of polymorphism ,virtual functions,inheritance and exception handling. CO 4. To understand file handling concepts using C++.	
8	**Course Outcomes (OC):** OC 1. Students can explain the key concept of OOP and their application in software development. OC 2. Students can Design and implement classes and objects to model real-world entities. OC 3. Students can apply the concepts of polymorphism, virtual functions, inheritance and exception handling in program. OC 4. Students can apply operator overloading, runtime polymorphism, generic Programming OC 5. Students can implement file handling concepts in program	
9	**Modules:-** **Module 1:**	
	1. Object Oriented Methodology: Introduction, Advantages and Disadvantages of Procedure Oriented Languages, Application of OOPS, Principles of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing. 2. Classes and Objects: Simple classes (Class specification, class members accessing), Defining member functions, passing object as an argument, Returning object from functions, friend classes, friend function.	**15 Hrs**

	3. Constructors and Destructors: Introduction, Default Constructor, Parameterized Constructor and examples, Destructors. 4. Program development using Inheritance: Introduction, Advantages provided by inheritance, choosing the access specifier, Derived class declaration, derived class constructors, class hierarchies, multiple inheritance, multilevel inheritance, hybrid inheritance.	
	Module 2:	
	5. Polymorphism: Concept of function overloading, overloaded operators, overloading unary and binary operators. 6. Virtual Functions: Introduction and need, Pure Virtual Functions, this Pointer, abstract classes, virtual destructors. 7. Exception Handling: Introduction, Exception Handling Mechanism, Concept of throw & catch with example. 8. Working with Files: Introduction, File Operations, Various File Modes, File Pointer and their Manipulation.	**15 Hrs**
10	**Text Books** 1. Object-oriented Programming C++, Hari Mohan Pandey , Laxmi Publications 2. C++ Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg , McGraw-Hill Education 3. C++ How to Program , Paul Deitel, Harvey Deitel	
11	**Reference Books** 1. Object Oriented Programming in C++ , E Balagurusamy 2. Object-Oriented Programming in C++ , Robert Lafore, Pearson Education. 3. Programming with ANSI C++ , Bhushan Trivedi 4. Demystified Object- Oriented Programming with C++, Dorothy R. Kirk	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper:** **External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Web Designing

Sr.No	Heading	Particulars
1	**Description the course : Including but Not limited to:**	The objective of Web Designing course is to provide instructions on creating and maintaining a web page for publishing on the Internet. Students will be able to use HTML editor to author pages that include text and graphics..
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the fundamentals of Internet, and the principles of web design CO 2. To develop basic websites using HTML and Style Sheets. CO 3. To understand different style sheets used in web designing. CO 4. To implement JavaScript as a tool to add dynamism to static HTML pages.	
8	**Course Outcomes (OC):** OC 1. Learners will be able to use the HTML programming language OC 2. Learners will be able to execute web pages designed using HTML OC 3. Describe the concepts of World Wide Web, and the requirements of effective web design OC 4**.** List various tags in html and **use** these to **create** web page OC 5 : Gain necessary skills for designing and developing web applications	
9	**Modules:-** **Module 1:**	
	1. Introduction to HTML 5: What Is HTML? Understanding HTML Tags, Setting Up the Document Structure: Specifying the Document Type, Creating the HTML, Specifying a Page Title. Formatting Text by Using Tags: Creating Headings, Applying Bold and Italic Formatting, Applying Superscript and Subscript Formatting, Using Monospace and Preformatted Text. Using Lists and Backgrounds: Creating Bulleted and Numbered Lists, Creating Definition Lists, Inserting Special Characters, Inserting Horizontal Lines, Choosing Background and Foreground Colors. Creating Hyperlinks and Anchors- Hyperlinking to a Web Page, Creating Hyperlinking to an E-Mail Address, Hyperlinking to Other Content. **Style Sheets and Graphics**: Introduction to Style Sheets: Understanding Styles, Constructing Style Rules, Creating Styles for Nested Tags, Applying Styles to Hyperlinks, Creating and Linking to External Style Sheets.	**15 Hrs**

		Formatting Text by Using Style Sheets: Specifying a Font Family, Specifying a Font Size and Color, Applying Bold and Italics, Applying Strikethrough and Underlining, Creating Inline Spans, Adjusting Spacing Between Letters. Formatting Paragraphs by Using Style Sheets: Indenting Paragraphs, Applying a Border to a Paragraph, Specifying the Horizontal Alignment of a Paragraph, **Displaying Graphics** Selecting a Graphics Format, Preparing Graphics for Web Use, Inserting Graphics, Arranging Elements on the Page, Controlling Image Size and Padding, Hyperlinking from Graphics, Using Thumbnail Graphics, Including Alternate Text for Graphics, Adding Figure Captions 2. Page Layout and Navigation- Creating Navigational Aids , Creating a Text-Based and Graphical Navigation Bar, Creating an Image Map, Creating Tables, Specifying the Size of a Table, Specifying the Width of a Column, Merging Table Cells. Formatting Tables-Applying Table Borders, Applying Borders by Using Attributes, Applying Borders by Using Styles, Changing Cell Padding, Spacing, and Alignment. Setting Horizontal and Vertical Alignment **Creating User Forms**- Creating a Basic Form- Creating a Text Box,Special Field types for E-Mail and Web Addresses, Creating a Text Area, Creating a Submit or Clear Button, Creating Check Boxes and Option Buttons, Additional Input Types in HTML5 Incorporating Sound and Video- What's New with Audio and Video in HTML5?,Embedding Video Clips- Introducing the <video> Tag, The <embed> Tag: Your Fallback Plan, Placing a Video Clip on a Web Page. Incorporating Audio on a Web Page- Playing Audio with the <audio> Tag, Placing an Audio Clip on a Web Page	
	Module 2:		
		1. JavaScript: Introduction to JavaScript: Variable, statements, Operators, Comments, constructs, Functions, expressions, JavaScript console, Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date,Methods,Arrays, Array Methods, Booleans, Comparisons, Control Structures: Conditions, Switch, Loop For, Loop While, Break. **Operators**: Arithmetic Operators, Assignment Operators, Comparison Operators, Logical Operators, Bitwise Operators Statements: Conditional Statements – if else, switch, Loops – while, do while, for, for in, for of, Loop Control – break, continue, labels JavaScript Objects: User-defined Objects, with Keyword, Native Objects – Array, String, Date, Math, Number, RegExp, Cookies Events and Event Handlers: HTML Events, DOM Events, DOM Event Listener,onAbort, onBlur, onChange, onClick, onDblClick, onError, onFocus, onKeyDown,onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut,onMouseOver, onMouseUp, onReset, onResize, onSelect, onSubmit, onUnload 2. Basics of JQuery, JQuery selection and events, JQuery Effects, JQuery traversal and manipulation, Data attributes and templates, jQuery Plugins.	**15 Hrs**

	3. JSON – JSON: Introduction, JSON Grammar, JSON Values, JSON Tokens, Syntax, JSON vs. XML, Data Types, Objects, Arrays, Creating JSON, JSON Object, Parsing JSON, Persisting JSON, Data Interchange, JSON HTML, JSONP	
10	**Text Books** 1. Step by Step HTML5 by Faithe Wempen, Microsoft Press,2011 2. The Complete Reference HTML & CSS, Thomas A. Powell. McGrawHill, 5 th Edition,2010 3. The Complete Reference JavaScript Thomas A. Powell &Fritz Schneider McGrawHill 3rd 2012 4. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Black Book Kindle Edition,by Kogent Learning Solutions Inc 5. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed Kindle Edition,by DT Editorial Services 6. JSON at work ,Tom MArrs,O'REILLY,First edition,2017	
11	**Reference Books** 1. Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics, Jennifer Niederst Robbins, O'Reilly, 5th Edition,2018. 2. Ivan Bayross, "Web Enabled Commercial Applications Development using HTML, DHTML, Javascript, Perl CGI", BPB, 2004 3. HTML 5 for Web Designers (By: Jeremy Keith) – http:// freepdf-books.com 4. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON kindle Edition by Lindsay Bassett, O'REILLY	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Major Practical II

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	**Object Oriented Programming usng C++ Practical** OOP encourages modular objects for reusable code, ensures well-organized and maintainable code via encapsulation, inheritance, and polymorphism, allowing flexibility and easy updates. Additionally, OOP models real-world scenarios, enhancing system understanding. **Web Designing Practical** Applying basic programming principles to the construction of websites

2	**Vertical :**	Major Practical
3	**Type :**	Practical
4	**Credits :**	2 credits (Total 60 hrs; 1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To explain the important characteristics of the C++ programming language. CO 2. To combine components of the C++ programming language to develop structured program. CO 3. To demonstrate the skills essential to compile, debug, and test C++ programs correctly. CO 4. To understand how to effectively implement HTML. CO 5. To develop the concept of basic and advanced text formatting. CO 6. To understand Hyper linking, Designing of webpage.	
8	**Course Outcomes (OC):** OC 1. Utilize C++ characteristics in software design and development. OC 2. Explain object-oriented techniques and explain how C++ supports them. OC 3. Employ C++ to demonstrate practical skill developing object-oriented solutions. OC 4. Examine a problem statements and design and develop object-oriented software using good coding practices and procedures. OC 5. Design static web pages using Hyper Text Markup Language (HTML). OC 6. Use their learned skills, knowledge and abilities to develop web sites OC 7. Collect information from the user with HTML Forms. OC 8. Enhance the look of web pages by implementing audio and video	
9	**Module I**	
	1. a. Write a C++ program to create a simple calculator. b. Write a C++ program to convert seconds into hours, minutes and seconds. c. Write a C++ program to find the volume of a square, cone, and rectangle. 2. a. Write a C++ program to find the greatest of three numbers. b. Write a C++ program to find the sum of even and odd n natural numbers c. Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user 3. a. Write a C++ program using classes and object Student to print name of the student, roll_no. Display the same. b. Write a C++ program for Structure bank employee to print name of the employee, account_no. & balance. Display the same also display the balance after withdraw and deposit	**30 Hrs**

c. Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not. WherereadNo() will be private method.

d. Write a program to demonstrate function definition outside class and accessing class members in function definition.

4.

a. Write a friend function for adding the two complex numbers, using a single class

b. Write a friend function for adding the two different distances and display its sum, using two classes.

c. Write a friend function for adding the two matrix from two different classes and display itssum

d. Write a Program to find Maximum out of Two Numbers using friend function.

Note: Here one number is a member of one class and the other number is member of some other class.

5.

a. Design a class Complex for adding the two complex numbers and also show the use of constructor.

b. Design a class Geometry containing the methods area() and volume() and also overload the area()function

c. Design a class StaticDemo to show the implementation of static variable and staticfunction

d. Write a C++ program to overload new/delete operators in a class.

e. Write a C++ Program to generate Fibonacci Series by using Constructor to initialize the Data Members.

6.

a. Overload the operator unary(-) for demonstrating operator overloading

b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.

c. Overload the + for concatenating the two strings. For e.g “Py” +“thon” =Python

7.

a. Implement the concept of method overriding.

b. Show the use of virtual function

c. Show the implementation of abstract class.

8.

a. Write a C++ Program that illustrate single inheritance.

b. Write a C++ Program that illustrate multiple inheritance.

c. Write a C++ Program that illustrate multi-level inheritance.

d. Write a C++ Program that illustrate Hierarchical inheritance.

9.

a. Show the implementation of exception handling

b. Show the implementation for exception handling for strings
c. Show the implementation of exception handling for using the pointers.

10.

a. Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.
b. Design a class to handle multiple files and file operations
c. Design a editor for appending and editing the files

Module II

1 Use of Basic and Advanced Tags, Lists and Backgrounds

a. Understanding elements, Tags and basic structure of HTML files
b. Design a web page using basic and advanced text formatting tags.
c. Design a web page using ordered, unordered list and description list.
d. Design a web page by choosing Background and Foreground Colors
e. Design a web page using Nested list and special characters.
f. Write an HTML code to display your CV on a web page.

2 Creating Hyperlinks, Anchors and style sheets

a. Design a web page with links to different pages and allow navigation between web pages.
b. Design a web page that automatically redirects the user to Other Content
c. Creating Hyperlinking to an E-Mail Address
d. Design a web page for creating Styles for Nested Tags
e. Design a web page by applying Styles to Hyperlinks
f. Design a web page by Creating and Linking to External Style Sheets.

3 Formatting Text and Paragraph by Using Style Sheets and displaying graphics

a. Design a web page by using text formatting tags
b. Design a web page using Indenting Paragraphs, Applying Border to a Paragraph and Specifying Horizontal Alignment of a Paragraph
c Implement a web page by creating inline spans and adjusting space between lines
d. Implement a web page by inserting a image and controlling the image size and padding
e. Design a web page by making image as a hyperlink
f. Develop a web page by using thumbnail graphics and also implement text for graphics

4 Tables , Page Layout and Navigation

a. Display a time table and display it in tabular format,
b. Write an html program to get the following output

NAME	SUBJECT	MARKS

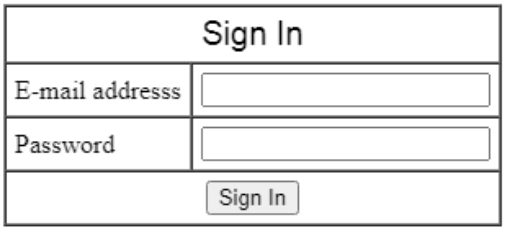
30 Hrs

Hillary	Advanced Web	75
	Operating System	60
Lary	Advanced Web	80
	Operating System	75
	Total Average: 72.5	

c. Design a table by merging the table cells.
d. Design a web page by Creating a Text-Based Navigation Bar
e . Design a web page by Creating a Graphical Navigation Bar
f. Design a web page with Image Map

5. Forms and Introducing video and audio tags

a. Design a web page with a form that uses all types of controls.
b. Design an admission form for any course in your college with text, pass word fields, check boxes, radio button and reset button.
c. Write a program to get the following output



d. Design a web page by placing a Video Clip on a Web Page
e. Design a web page by placing an Audio Clip on a Web Page
f. Design a web page embedding image, audio and video.

6 .Basics of java script
a. Using JavaScript, design a web page to accept a number from the user and print its Factorial.
b. Using JavaScript, a web page that prints Fibonacci series/any given series.
c. Write a JavaScript program to display all the prime numbers between 1 and 100.
d. Write a JavaScript program to accept a number from the user and display the sum of its digits.

7. Java Script: Validating User fields
a. Demonstrate the use of Document object methods.
b. Using java script, demonstrate validating Text Input Fields, Drop-down Lists and Checkboxes
c. Using java script, demonstrate validating Radio buttons and Validating Multi-Select Boxes
d. Write a Java script to prompt for users name and display it on the screen.

	8. Java Script : Handling the events a.Using java script, demonstrate the use of onAbort, onBlur, onChange, onClick, onDblClick events b. Using java script, demonstrate the use of onDragDrop, onError, onFocus events c. Using java script, demonstrate the use of onKeyDown, onKeyPress, onKeyUp, onLoad, onReset, onResize, onSelect, onSubmit, onUnload events d. Using java script, demonstrate the use of onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove events. e. Using java script, demonstrate the use of onKeyDown, onKeyPress, onKeyUp, onLoad, onReset, onResize, onSelect, onSubmit, onUnload events 9. JQuery a. use JQuery effect in page b. Write a jQuery Code to find the data passed with the on() method for each element. c. Use JQuery Events d. JQuery traversal and manipulation 10. JSON Basics and Working with JSON a. Creating JSON b. Parsing JSON c. Persisting JSON d. Demonstrate use of JSON objects in array, print array on web page using document object e. Read data from json file and convert it into a JavaScript object and display the data in web page using document object	
10	Text Books 1. Object-oriented Programming C++, Hari Mohan Pandey 2. C++ Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg 3. C++ How to Program, Paul Deitel, Harvey Deitel 4. Step by Step HTML5, Faithe Wempen, Microsoft Press,2011 5. The Complete Reference HTML & CSS, Thomas A. Powell. McGraw Hill, 5th Edition,2010	
11	**Reference Books** 1. Object Oriented Programming in C++, E Balagurusamy 2. Object-Oriented Programming in C++ by Robert Lafore 3. Programming with ANSI C++, Bhushan Trivedi 4. Demystified Object- Oriented Programming with C++, Dorothy R. Kirk 5. Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics, Jennifer Niederst Robbins, O'Reilly, 5th Edition,2018. 6. "Web Enabled Commercial Applications Development using HTML, DHTML, Javascript, Perl CGI", Ivan Bayross, BPB, 2004 7. HTML 5 for Web Designers (By: Jeremy Keith) – http:// freepdf-books.com	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Vocational Skill Courses (VSC)

Name of the Course: Assembly Language Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Introduction: The 8085 Assembly Language Programming course covers the principles and practices of writing low-level software that controls the 8085 microprocessor. This course provides an in-depth understanding of the 8085 microprocessor architecture and its instruction set, as well as how to write, debug, and optimize assembly language programs for this microprocessor. Relevance and Usefulness: The course is relevant to computer science/engineering students interested in learning about microprocessors and embedded systems programming. The course provides the fundamental knowledge and skills required to design and implement computer systems with low-level software control. Assembly language programming is the foundation of modern computer technology, which makes the course relevant to anyone interested in computer systems and programming. Application and Interest: The course is essential for students aspiring to work in the field of embedded systems, microcontroller/microprocessor programming, or any programming role that involves low-level software development. By the end of the course, students will be able to write efficient and optimized assembly language programs that control the functionality of a microprocessor. Connection with Other Courses: 8085 Assembly Language Programming is a fundamental course that provides an understanding of how computer systems work at the lowest level. It connects with several other computer science courses, such as Computer Organization and Architecture, Operating Systems, Compiler Design, and Embedded Systems Design. Demand in the Industry and Job Prospects: There is a high demand in the industry for programmers who possess knowledge of low-level software development, such as programming microprocessors with assembly language. Many industries, including aerospace, automotive, healthcare, and consumer electronics, require low-level software development skills in their employees. Job prospects for graduates with expertise in 8085 Assembly language

		programming are abundant in these sectors. Job roles may include embedded software engineer, hardware engineer, firmware developer, software developer, and testing/validation engineer.
2	**Vertical :**	Vocational Skill Course(VSC)
3	**Type :**	Practical
4	**Credits :**	2 credits (60 hours in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To gain a thorough understanding of the 8085 microprocessor architecture and its associated instruction set. CO 2. To develop the ability to write and debug assembly language programs for the 8085 microprocessor. CO 3. To learn the principles of computer organization and how they relate to the 8085 microprocessor. CO 4. To become proficient in the use of 8085 assembly language programming tools, simulators, and debuggers. CO 5. To learn how to interface different input/output devices with the 8085 microprocessor. CO 6. To understand the concept of interrupts and how they are used in 8085 assembly language programming.	
8	**Course Outcomes(CO):** OC 1. Explain the architecture of the 8085 microprocessor and its associated instruction set. OC 2. Identify the different types of registers and their functions in the microprocessor. OC 3. Describe the memory organization and addressing modes of the 8085 microprocessor. OC 4. Write assembly language programs for the 8085 microprocessor using various instructions and addressing modes. OC 5. Debug and troubleshoot assembly language programs for the 8085 microprocessor using simulators and debuggers. OC 6. Implement conditional branching and looping constructs in assembly language programs. OC 7. Use 8085 assembly language programming tools, such as editors, assemblers, and emulators for developing and testing programs. OC 8. Simulate microprocessor operations using emulators and debuggers. OC 9. Connect input/output devices, such as LEDs, switches, and displays, to the 8085 microprocessor. OC 10.	
9	**Modules:-** **Module 1:**	
	1. Perform the following Operations related to memory locations. a. Store the data byte 32H into memory location 4000H. b. Exchange the contents of memory locations 2000H and 4000H **2. Simple assembly language programs.**	**30 Hrs**

a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
b. Subtract two 8-bit numbers.
c. Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
d. Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.
e. Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
f. Find the l's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.
g. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.

3. Packing and unpacking operations.

a. Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.
b. Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit.

4. Register Operations

a. Write a program to shift an eight bit data four bits right. Assume that data is in register C.
b. Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair
c. Write a set of instructions to alter the contents of flag register in 8085.
d. Write a program to count number of I's in the contents of D register and store the count in the B register.

5. Multiple memory locations.

a. Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H
b. Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.
c. Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.

	d. Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H e. Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.	
	Module 2:	
	1. Calculations with respect to memory locations. a. Write a program to sort given 10 numbers from memory location 2200H in the ascending order. b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2Sample problem: c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H. d. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H e. Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H f. Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H g. Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array I, array2 and array3 are 2200H, 2300H and 2400H, respectively **2. Assembly programs on memory locations.** a. Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H b. Write assembly language program with proper comments for the following: c. A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.	**30 Hrs**

	d. Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location 2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character. e. A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively f. Write an assembly language program to generate Fibonacci number. g. Program to calculate the factorial of a number between 0 to 8. **3. String operations in assembly programs.** a. Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters b. Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters. c. Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H. d. Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H e. DAA instruction is not present. Write a sub routine which will perform the same task as DAA. **4. Calculations on memory locations.** a. To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10 b. Arrange an array of 8 bit unsigned no in descending order c. Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H d. Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H. e. Write a simple program to Split a HEX data into two nibbles and store it in memory **5. Operations on BCD numbers.** a. Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit. b. Subtract the BCD number stored in E register from the number stored in the D register c. Write an assembly language program to multiply 2 BCD numbers	
10	**Text Books** 1. 8080A/8085 Assembly Language Programming, Lance A. Leventhel, Osborne, 1978	

11	**Reference Books** 1. Microprocessors Architecture, Programming and Applications with the 8085, Fifth Edition, Penram Publications, 2012	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Skill Enhancement Courses (SEC)

Name of the course : Web Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course covers a range of topics aimed at equipping students with the skills and knowledge needed to create visually appealing, functional, and user-friendly websites. The course provides an insight into emerging technologies to design and develop state of the art web applications using client-side scripting, server-side scripting, and database connectivity. website development includes all related development tasks, such as client-side scripting, server-side scripting, server and network security configuration, eCommerce development, and content management system (CMS) development. Website design is a combination of different elements that work together to create an effective and user-friendly experience. These include the use of typography, layout, color theory, grid systems, motion graphics, and responsive designs.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO)** **CO1:** To understand how to use Java script objects and XML. **CO2:** To create well organized, styled web pages **CO3:** To add versatility to a web page using jQuery **CO4:** To deploy a local web server and run a simple web application. **CO5:** To read and process data in MySQL using PHP. **CO6:** To understand usage of Bootstrap	
8	**Course Outcomes (OC)** OC1: Knowledge in different java script objects. OC2: How to use XML with CSS and XSL OC3: validate a form using jQuery OC4: handle asynchronous requests OC4: Write and deploy PHP with database and to simplify web development. OC5: Create a responsive layout using the Bootstrap	

	9	**Modules:** **Module 1:** **1. Write JavaScript code for** a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document c. Storing and Retrieving Cookies **2. Create a XML file with Internal / External DTD and display it using** a. CSS b. XSL **3. Write PHP scripts for- Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions** **4. Write PHP scripts for** a. Retrieving data from HTML forms b. Working with Arrays c. Working with Files (Reading / Writing) **5. Advanced PHP** a. Write a PHP program to demonstrate use of sessions and cookies. b. Write a PHP program to demonstrate use of filters.	**30 Hrs**
		Module 2 **6. PHP and MySQL** a. Write a PHP program to create: Create a database College b. Create a table Department (Dname, Dno, Number_of_faculty) c. Write a PHP program to create a database named “College”. Create a table named “Student” with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format. **7. Write a PHP program** a. Update rows in a table b. Delete rows from a table **8. Design a PHP page for authenticating a user** **9. Write PHP scripts for** a. Storing and Retrieving Cookies b. Storing and Retrieving Sessions **10. Perform the following using Bootstrap:** a. Create a responsive layout using the Bootstrap grid system b. Create a simple Bootstrap navbar with dropdown menus c. Create a basic Bootstrap form with validation	**30 Hrs**
	10	**Text Books** • HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press, 2016 • Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India, 2018	

	• PHP: A Beginners Guide, Vikram Vaswani, TMH	
11	**Reference Books** • HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY, 2011 • Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly, 2018 • PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley, 2018	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Name of the Course: PLSQL Practical

Sr.No.	Heading	Particulars	
1	**Description the course : Including but Not limited to:**	PL/SQL ,Oracle's procedural extension language for SQL, allows developers to include procedural language components such as loops, conditional statements and functions. The course enables students with practical experience in using PL/SQL for effective database programming and development.	
2	**Vertical :**	Skill Enhancement Course(SEC)	
3	**Type :**	Practical	
4	**Credits :**	2 credits	
5	**Hours Allotted :**	60 Hours	
6	**Marks Allotted:**	50 Marks	
7	**Course Objectives(CO):** CO 1. Comprehend the basics of PL/SQL and gain knowledge about control and conditional statement in PL/SQL. CO 2. Understand working with cursors,collections and composite data types in PL/SQL. CO 3. Develop expertise in creating stored procedures and functions. CO 4. Explore the use of triggers to automate responses to events within the database. CO 5. Understand the concept of Exception handling. CO 6. Design modular applications using packages.		
8	**Course Outcomes (OC):** OC 1. Use PL/SQL variables ,data types, control and conditional statement. OC 2. Apply sequences and cursor in PL/SQL. OC 3. Work with Collection and Composite Data Types. OC 4. Develop PL/SQL structures like functions, procedures and triggers for database applications. OC 5. Handle errors and exceptions in PL/SQL programs. OC 6. Develop PL/SQL packages.		
9	**Modules:-** **Module 1:**		
	1. **PL/SQL Basics**- Use of variables, Write executable statement, Interacting with Oracle Server, Create anonymous PL/SQL block,Sequences 2. **Control Structure in PL/SQL**- Using while loop, Do loop, For loop, Use of GOTO statement 3. **Create conditional statement using PL/SQL**- Using if statement, Using if else statement, Using elsif ladder, Using case expression. 4. **Create cursor in PL/SQL**- Implicit cursor, Explicit cursor, Parameterized cursor **5. Collection and Composite Data Types** - Working with Collections,Working with Composite Data Types		**30 Hrs**

	Module 2:	
	1. Creation of Procedures in PL/SQL **2. Functions in PL/SQL** 3. **Creation of Trigger –** Create Row level trigger, Create Statement level trigger, Create instead of trigger 4. **Handling exceptions**- Creation of user defined exception, Creation of system defined exception 5. **Creation of Package in PL/SQL**	**30 Hrs**
10	**Text Books** 1. Programming with PL/SQL for Beginners , H. Dand, R. Patil and T. Sambare, X –Team 2. Oracle pl/sql Programming ,Feuerstein, S., & Pribyl, B. ," O'Reilly Media, Inc.".	
11	**Reference Books** 1. Oracle Database PL/SQL Language Reference, 12c Release 1 (12.1) E50727-04 , Alpern, D., Belden, E., Agrawal, S., Baer, H., Castledine, S., Chang, T., & Yang, M. 2. Oracle PL/SQL for dummies , Rosenblum, M., & Dorsey, P. (2006), John Wiley & Sons. 3. PL/SQL Programming ,Ivan Bayross, BPB	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

QUESTION PAPER PATTERN

(External and Internal)

<table>
<tr><td rowspan="4">I</td><td colspan="2">A Theory of 2 credits is evaluated for a total of 50 Marks</td></tr>
<tr><td>Internal Continuous Assessment:</td><td>40%[20 Marks]</td></tr>
<tr><td colspan="2">Continuous Evaluation through:
Class test of 1 of 15 marks
Class test of 2 of 15 marks
Average of the two: 15 marks

Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td></tr>
<tr><td colspan="2">External Semester End Examination: 60%[30 Marks]</td></tr>
<tr><td colspan="3">Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)
Q1: Attempt any two (out of four) from Module 1 (15 marks)
Q2: Attempt any two (out of four) from Module 2 (15 marks)</td></tr>
<tr><td rowspan="4">II</td><td colspan="2">A Practical of 2 credits is evaluated for a total of 50 Marks</td></tr>
<tr><td>Internal Continuous Assessment:</td><td>40%[20 Mrks]</td></tr>
<tr><td colspan="2">Continuous Evaluation through:
Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.</td></tr>
<tr><td colspan="2">Semester End Examination: 60%[30 Marks]</td></tr>
<tr><td colspan="3">Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination(30 Marks)
Practical Slip:
Q1. From Module 1 13 marks
Q2. From Module 2 12marks
Q3. Journal and Viva 05 marks</td></tr>
</table>

Examination and Standard of Passing:

Regulations regarding the scheme of exams, number of credits and standard of passing will be as prescribed by the University of Mumbai.

A student is said to have passed if he/she secures 40% of marks allotted in each head of passing. External evaluation of 30 marks and Internal evaluation of 20 marks are treated as separate heads of passing.

The Ten Point Grading System prescribed by the University of Mumbai will be as follows:

Letter Grades and Grade Points

Semester GPA/ Program CGPA Semester/ Program	**% of Marks**	**Alpha-Sign / Letter GradeResult**	**Grade Points**
9.00-10.00	90.0-100	O (Outstanding)	10
8.00-<9.00	80.0-<90.0	A+ (Excellent)	9
7.00-<8.00	70.0-<80.0	A (Very Good)	8
6.00-<7.00	60.0-<70.0	B+ (Good)	7
5.50-<6.00	55.0-<60.0	B (Above Average)	6
5.00-<5.50	50.0-<55.0	C (Average)	5
4.00-<5.00	40.0-<50.0	P (Pass)	4
Below 4.00	Below 40	F (Fail)	0
Ab (Absent)	-	Absent	0

This syllabus is applicable to IDOL students as well, w.e.f. 2025-26

Appendix B

Justification for B.Sc. (Information Technology)

1.	Necessity for starting the course:	**A large amount of The demand for IT professionals is consistently high, and individuals with a B.Sc in IT can find opportunities in various sectors, including technology companies, healthcare, finance, government, and more.**
2.	Whether the UGC has recommended the course:	**Yes**
3.	Whether all the courses have commenced from the academic year 2024-2025	**To be implemented from 2024-2025 onwards**
4.	The courses started by the University are self-financed, whether adequate number of eligible permanent faculties are available?:	**Self-financed** **Yes. Some experts are called as visiting faculties**
5.	To give details regarding the duration of the Course and is it possible to compress the course?:	**4 years. Not possible to compress the program**
6.	The intake capacity of each course and no. of admissions given in the current academic year:	**60 seats for one division. Admissions will be held from 2024-2025 onwards**
7.	Opportunities of Employability / Employment available after undertaking these courses:	**B.Sc in Information Technology can open up a wide range of opportunities and employment prospects across various industries. Additionally, as technology continues to advance, new roles and specialties within the IT field are continually emerging, providing diverse career paths for IT graduates.**

Sign of Chairperson
Dr. Mrs. R. Srivaramangai
Ad-hoc BoS (IT)

Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sign of Offg. Dean,
Prof. Shivram S. Garje
Faculty of Science & Technology