Scheme of Courses for S.Y.B.Sc. I.T.

Semester – III				
Course Code	**Course Type**	**Course Title**	**Credits**	**No of lects/ week**
USIT301	Skill Enhancement Course	Python Programming	2	5
USIT302	Core Subject	Data Structures	2	5
USIT303	Core Subject	Computer Networks	2	5
USIT304	Core Subject	Operating Systems	2	5
USIT305	Core Subject	Applied Mathematics	2	5
USIT3P1	Skill Enhancement Course Practical	Python Programming Practical	2	3
USIT3P2	Core Subject Practical	Data Structures Practical	2	3
USIT3P3	Core Subject Practical	Computer Networks Practical	2	3
USIT3P4	Core Subject Practical	Operating Systems Practical	2	3
USIT3P5	Core Subject Practical	Mobile Programming Practical	2	3
		Total Credits	**20**	

Semester – IV				
Course Code	**Course Type**	**Course Title**	**Credits**	**No of lects/ week**
USIT401	Skill Enhancement Course	Core Java	2	5
USIT402	Core Subject	Introduction to Embedded Systems	2	5
USIT403	Core Subject	Computer Oriented Statistical Techniques	2	5
USIT404	Core Subject	Software Engineering	2	5
USIT405	Core Subject	Computer Graphics and Animation	2	5
USIT4P1	Skill Enhancement Course Practical	Core Java Practical	2	3
USIT4P2	Core Subject Practical	Introduction to Embedded Systems Practical	2	3
USIT4P3	Core Subject Practical	Computer Oriented Statistical Techniques Practical	2	3
USIT4P4	Core Subject Practical	Software Engineering Practical	2	3
USIT4P5	Core Subject Practical	Computer Graphics and Animation Practical	2	3
		Total Credits	**20**	

SEMESTER III

Syllabus for S. Y. B. Sc. I.T. Autonomous for the year 2023-24

Name of the Course	**Python Programming**
Course Code (refer to student handbook)	**USIT301**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	**Theory**
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	In the Python Programming course, the emphasis is on fostering employability, entrepreneurship, and skill development. Through hands-on projects and real-world applications, students gain practical programming expertise sought after by the job market. They learn to develop software solutions, enhancing their employability prospects in various industries. Additionally, the course encourages innovative thinking and problem-solving, nurturing an entrepreneurial mindset among students. They acquire coding proficiency, making them well-equipped to tackle challenges and explore entrepreneurial ventures in the technology domain. By honing their Python programming skills, students become valuable assets to potential employers and are empowered to create their own opportunities in the ever-evolving world of IT and business.

Nomenclature: Python Programming

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Aware of the variables, expressions, looping and conditions used in Python programming.
2. Implement functions, strings, lists, tuples and directories
3. Create GUI forms and add widgets.
4. Use MySQL to store data.
5. Apply the programming skill set learnt here into various domains by having advance programming skill set of Python and usage of libraries.

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**Introduction**	The Python Programming Language, History, features, Installing Python, Running	**12**
	Python program, Debugging	Syntax Errors, Runtime Errors, Semantic Errors, Experimental Debugging, Formal and Natural Languages, The Difference Between Brackets, Braces, and Parentheses	
	Variables and Expressions	Values and Types, Variables, Variable Names and Keywords, Type conversion, Operators and Operands, Expressions, Interactive Mode and Script Mode, Order of Operations.	
	Conditional Statements	if, if-else, nested if –else	
	Looping	for, while, nested loops Control statements: Terminating loops, skipping specific conditions	
II	**Functions**	Function Calls, Type Conversion Functions, Math Functions, Composition, Adding New Functions, Definitions and Uses, Flow of Execution, Parameters and Arguments, Variables and Parameters Are Local, Stack Diagrams, Fruitful Functions and Void Functions, Why Functions? Importing with from, Return Values, Incremental Development, Composition, Boolean	**12**

		Functions, More Recursion, Leap of Faith, Checking Types	
	Strings	A String Is a Sequence, Traversal with a for Loop, String Slices, Strings Are Immutable, Searching, Looping and Counting, String Methods, The in Operator, String Comparison, String Operations.	
III	**Lists**	Values and Accessing Elements, Lists are mutable, traversing a List, Deleting elements from List, Built-in List Operators, Concatenation, Repetition, In Operator, Built-in List functions and methods	**12**
	Tuples and Dictionaries	Tuples, Accessing values in Tuples, Tuple Assignment, Tuples as return values, Variable-length argument tuples, Basic tuples operations, Concatenation, Repetition,in Operator, Iteration, Built-in Tuple Functions Creating a Dictionary, Accessing Values in a dictionary, Updating Dictionary, Deleting Elements from Dictionary, Properties of Dictionary keys, Operations in Dictionary, Built-In Dictionary Functions, Built-in Dictionary Methods	
	Files	Text Files, The File Object Attributes, Directories	
	Exceptions	Built-in Exceptions, Handling Exceptions, Exception with Arguments, User-defined Exceptions	
IV	**Regular Expressions**	Concept of regular expression, various types of regular expressions, using match function.	**12**
	Classes and Objects	Overview of OOP (Object Oriented Programming), Class Definition, Creating Objects, Instances as Arguments, Instances as return values, Built-in Class Attributes, Inheritance, Method Overriding, Data Encapsulation, Data Hiding	
	Multithreaded Programming	Thread Module, creating a thread, synchronizing threads, multithreaded priority queue	

	Modules	Importing module, Creating and exploring modules, Math module, Random module, Time module	
V	**Creating the GUI Form and Adding Widgets: Widgets**	Widgets: Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Message, Radiobutton, Scale, Scrollbar, text, Toplevel, Spinbox, PanedWindow, LabelFrame, tkMessagebox. Handling Standard attributes and Properties of Widgets.	**12**
	Layout Management	Designing GUI applications with proper Layout Management features.	
	Look and Feel Customization	Enhancing Look and Feel of GUI using different appearances of widgets, Storing Data in Our MySQL	
	Database via Our GUI	Connecting to a MySQL database from Python, Configuring the MySQL connection, Designing the Python GUI database, Using the INSERT command, Using the UPDATE command, Using the DELETE command, Storing and retrieving data from MySQL database.	

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Think Python	Allen Downey	O'Reilly	1st	2012
2	An Introduction to Computer Science using Python 3	Jason Montojo, Jennifer Campbell, Paul Gries	SPD	1st	2014
3	Python GUI Programming Cookbook	Burkhard A. Meier	Packt		2015
4	Introduction to Problem Solving with Python	E. Balagurusamy	TMH	1st	2016
5	Murach's Python programming	Joel Murach, Michael Urban	SPD	1st	2017
6	Object-oriented Programming in Python	Michael H. Goldwasser, David Letscher	Pearson Prentice Hall	1st	2008
7	Exploring Python	Budd	TMH	1st	2016

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	Marks
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	Unit	Marks
1	I	12
2	II	12
3	III	12
4	IV	12
5	V	12

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions, single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Python Programming Practical**
Course Code (refer to student handbook)	**USIT3P1**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Python Programming Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Demonstrate proficiency in Python programming concepts, syntax, and standard libraries through hands-on coding exercises and projects.(Proficiency in Python)
2. Apply Python programming skills to solve real-world problems, creating practical solutions that can be implemented in various domains.(Real-World Application)
3. Develop problem-solving and critical thinking skills by tackling challenging programming tasks and overcoming coding obstacles.(Problem-Solving Abilities)
4. Gain valuable hands-on experience through practical assignments, enhancing employability and making students more desirable to potential employers.(Practical Experience)
5. Foster an entrepreneurial mindset by encouraging creative thinking and innovation, empowering students to explore entrepreneurial ventures in the technology landscape.(Entrepreneurial Mindset)

Curriculum:

Sr. No.	**Title**	**Learning Points**	**No of Lectures**
1	**Practical 1**	Write the program for the following: a. Create 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. Enter the 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 generate the Fibonacci series. d. Write a function that reverses the user defined value. e. Write a function to check the input value is Armstrong and also write the function for Palindrome. f. Write a recursive function to print the factorial for a given number.	**3**
2	**Practical 2**	Write the program for the following: a. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.	**3**

		b. Define a function that computes the length of a given list or string. c. Define a procedure histogram() that takes a list of integers and prints a histogram to the screen. For example, histogram([4, 9, 7]) should print the following: **** ********* *******	
3	**Practical 3**	Write the program for the following: a. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not. b. Take a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5.	**3**
4	**Practical 4**	Write the program 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 program to print a specified list after removing the 0th, 2nd, 4th and 5th elements. c. Write a Python program to clone or copy a list	**3**
5	**Practical 5**	Write the program for the following: a. Write a Python script to sort (ascending and descending) a dictionary by value. b. Write a Python script to concatenate the following dictionaries to create a new one. Sample Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60} Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60} c. Write a Python program to sum all the items in a dictionary.	**3**
6	**Practical 6**	Write the program for the following: a. Write a Python program to read an entire text file. b. Write a Python program to append text to a file and display the text. c. Write a Python program to read last n lines of a file.	**3**

7	**Practical 7**	Write the program for the following: a. Design a class that store the information of student and display the same b. Implement the concept of inheritance using python c. Create a class called Numbers, which has a single class attribute called MULTIPLIER, and a constructor which takes the parameters x and y (these should all be numbers). i. Write a method called add which returns the sum of the attributes x and y. ii. Write a class method called multiply, which takes a single number parameter a and returns the product of a and MULTIPLIER. iii. Write a static method called subtract, which takes two number parameters, b and c, and returns b - c. iv. Write a method called value which returns a tuple containing the values of x and y. Make this method into a property, and write a setter and a deleter for manipulating the values of x and y.	**3**
8	**Practical 8**	Write the program for the following: a. Open a new file in IDLE (“New Window” in the “File” menu) and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the “Control Flow and Functions” exercise into this file and save it. Now open a new file and save it in the same directory. You should now be able to import your own module like this:import geometry Try and add print dir(geometry) to the file and run it. Now 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 Trueand 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. b. Write a program to implement exception handling.	**3**
9	**Practical 9**	Write the program for the following: a. Try to configure the widget with various options like: bg=”red”, family=”times”,size=18 b. Try to change the widget type and configuration options to experiment with other widget types like	**3**

		Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.	
10	**Practical 10**	Design the database applications for the following: a. Design a simple database application that stores the records and retrieve the same. b. Design a database application to search the specified record from the database. c. Design a database application to that allows the user to add, delete and modify the records.	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1	Think Python	Allen Downey	O’Reilly	1st	2012
2	An Introduction to Computer Science using Python 3	Jason Montojo, Jennifer Campbell, Paul Gries	SPD	1st	2014

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	**Marks**
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Data Structures**
Course Code (refer to student handbook)	**USIT302**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	In the Data Structures course, the focus is on enhancing employability, fostering entrepreneurship, and promoting skill development. Through a hands-on approach and practical implementations, students develop proficiency in various data structures, making them highly desirable to potential employers seeking skilled programmers and software developers. The course emphasizes problem-solving and algorithmic thinking, nurturing an entrepreneurial mindset among learners. By mastering data structures and their applications, students are equipped to tackle complex challenges in diverse industries, enhancing their employability prospects. Additionally, the acquired skills empower students to explore innovative solutions and venture into entrepreneurial endeavors, contributing to the technology-driven business landscape. Overall, this Data Structures course prepares students for success in their careers and entrepreneurial pursuits.

Nomenclature: Data Structures

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Identify and distinguish data structure classification, data types, their complexities
2. Understand array, linked list, stack and queue.
3. Understand trees, various hashing techniques and graph for various applications
4. Compare various sorting and searching techniques.
5. Choose appropriate algorithm design techniques for solving problems.

Curriculum:

Unit	**Title**	**Learning Points**	**No of Lectures**
I	**Introduction**	Data and Information, Data Structure, Classification of Data Structures, Primitive Data Types, Abstract Data Types, Data structure vs. File Organization, Operations on Data Structure, Algorithm, Importance of Algorithm Analysis, Complexity of an Algorithm, Asymptotic Analysis and Notations, Big O Notation, Big Omega Notation, Big Theta Notation, Rate of Growth and Big O Notation.	**12**
	Array	Introduction, One Dimensional Array, Memory Representation of One Dimensional Array, Traversing, Insertion, Deletion, Searching, Sorting, Merging of Arrays, Multidimensional Arrays, Memory Representation of Two Dimensional Arrays, General Multi-Dimensional Arrays, Sparse Arrays, Sparse Matrix, Memory Representation of Special kind of Matrices, Advantages and Limitations of Arrays.	

II	**Linked List**	Linked List, One-way Linked List, Traversal of Linked List, Searching, Memory Allocation and Deallocation, Insertion in Linked List, Deletion from Linked List, Copying a List into Another List, Merging Two Linked Lists, Splitting a List into Two Lists, Reversing One way linked List, Circular Linked List, Applications of Circular Linked List, Two way Linked List, Traversing a Two way Linked List, Searching in a Two way linked List, Insertion of an element in Two way Linked List, Deleting a node from Two way Linked List, Header Linked List, Applications of the Linked list, Representation of Polynomials, Storage of Sparse Arrays, Implementing other Data Structures.	**12**
III	**Stack**	Introduction, Operations on the Stack Memory Representation of Stack, Array Representation of Stack, Applications of Stack, Evaluation of Arithmetic Expression, Matching Parenthesis, infix and postfix operations, Recursion.	**12**
	Queue	Introduction, Queue, Operations on the Queue, Memory Representation of Queue, Array representation of queue, Linked List Representation of Queue, Circular Queue, Some special kinds of queues, Deque, Priority Queue, Application of Priority Queue, Applications of Queues.	
IV	**Sorting and Searching Techniques**	Bubble, Selection, Insertion, Merge Sort. Searching: Sequential, Binary, Indexed Sequential Searches, Binary Search.	**12**
	Tree	Tree, Binary Tree, Properties of Binary Tree, Memory Representation of Binary Tree, Operations Performed on Binary Tree, Reconstruction of Binary Tree from its Traversals, Huffman Algorithm, Binary Search Tree, Operations on Binary Search Tree, Heap, Memory Representation of Heap, Operation on Heap, Heap Sort.	
	Advanced Tree Structures	Red Black Tree, Operations Performed on Red Black Tree, AVL Tree, Operations performed on AVL Tree, 2-3 Tree, B-Tree.	

V	**Hashing Techniques**	Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and rehashing	**12**
	Graph	Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List or Linked Representation of Graph, Operations Performed on Graph, Graph Traversal, Applications of the Graph, Reachability, Shortest Path Problems, Spanning Trees.	

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1	A Simplified Approach to Data Structures	Lalit Goyal, Vishal Goyal, Pawan Kumar	SPD	1st	2014
2	An Introduction to Data Structure with Applications	Jean – Paul Tremblay and Paul Sorenson	Tata MacGraw Hill	2nd	2007
3	Data Structure and Algorithm	Maria Rukadikar	SPD	1st	2017
4	Schaum's Outlines Data structure	Seymour Lipschutz	Tata McGraw Hill	2nd	2005
5	Data structure – A Pseudocode Approach with C	AM Tanenbaum, Y Langsam and MJ Augustein	Prentice Hall India	2nd	2006
6	Data structure and Algorithm Analysis in C	Weiss, Mark Allen	Addison Wesley	1st	2006

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Data Structures Practical**
Course Code (refer to student handbook)	**USIT3P2**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Data Structures Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Implement fundamental data structures efficiently.
2. Analyze and compare data structures for performance.
3. Solve real-world problems using data structures and algorithms.
4. Optimize code and troubleshoot errors in implementations.

Curriculum:

Sr. No.	Title	Learning Points	No of Lectures
1	**Programs using Arrays**	1. Implement the following: a. Write a program to store the elements in 1-D array and perform the operations like searching, sorting and reversing the elements. [Menu Driven] b. Read the two arrays from the user and merge them and display the elements in sorted order.[Menu Driven] c. Write a program to perform the Matrix addition, Multiplication and Transpose Operation. [Menu Driven]	**3**
2	**Programs on Linked List**	Implement the following for Linked List: a. Write a program to create a singly linked list and display the node elements in reverse order. b. Write a program to search the elements in the linked list and display the same c. Write a program to create a doubly linked list and sort the elements in the linked list.	**3**
3	**Programs on Stack**	3. Implement the following for Stack: a. Write a program to implement the concept of Stack with Push, Pop, Display and Exit operations. b. Write a program to convert an infix expression to postfix and prefix conversion. c. Write a program to implement the Tower of Hanoi problem.	**3**
4	**Programs on Queue**	Implement the following for Queue: a. Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations.	**3**

		b. Write a program to implement the concept of Circular Queue c. Write a program to implement the concept of Deque.	
5	**Programs on Sorting**	Implement the following sorting techniques: a. Write a program to implement bubble sort. b. Write a program to implement selection sort. c. Write a program to implement insertion sort.	**3**
6	**Programs on Searching Techniques**	Implement the following data structure techniques: a. Write a program to implement merge sort. b. Write a program to search the element using sequential search. c. Write a program to search the element using binary search.	**3**
7	**Programs on Tree**	Implement the following data structure techniques: a. Write a program to create the tree and display the elements. b. Write a program to construct the binary tree. c. Write a program for inorder, postorder and preorder traversal of tree	**3**
8	**Programs on Advanced Tree Structures**	Implement the following data structure techniques: a. Write a program to insert the element into maximum heap. b. Write a program to insert the element into minimum heap.	**3**
9	**Programs on Hashing Techniques,**	Implement the following data structure techniques: a. Write a program to implement the collision technique. b. Write a program to implement the concept of linear probing.	**3**
10	**Programs on Graph**	Implement the following data structure techniques: a. Write a program to generate the adjacency matrix. b. Write a program for the shortest path diagram.	**3**

Note : 1. Learners may opt for any preferable languages from C/C++/Python/Java for implementation of practical programs.

2. This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Data Structures and Algorithms Using Python	Rance Necaise	Wiley	First	2016
2	Data Structures Using C and C++	Langsam,Augenstein Tanenbaum	Pearson	First	2015

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	Marks
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	Unit	Marks
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Computer Networks**
Course Code (refer to student handbook)	**USIT303**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Computer Networks

Course Outcomes:

On the successful completion of this course, the learner will be able to:

1. Identify various data communication standards, topologies and terminologies
2. Describe how signals are used to transfer data and communication aspects between nodes.
3. Configure IP addresses using TCP/IP protocol suite
4. Understand the Transport layer and Network layer protocols
5. Use different application layer protocols.

Curriculum:

Unit	Title	Learning Points	No of Lectures
1	**Introduction**	Computer Network, Evolution of Computer Networks Different types of Computer Network, Difference between LAN, MAN and WAN, Hardware Devices used for Networking: Network Interface Card(NIC), Modem, Hub, Switch L1 and L2 switches, Comparison between switch and hub, Bridge, Router, Gateway. Standards and administration. Network Models: Protocol layering, TCP/IP protocol suite, The OSI model.	**12**
2	**Introduction to Physical layer**	Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance.	**12**
	Introduction to the Data Link Layer	Link layer addressing, Data Link Layer Design Issues, Error detection and correction, block coding	
	Wireless LANs	Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks.	
3	**Network Layer**	IPv4 Addresses, IPv4 Protocol, ARP, ICMP, IPv6	**12**
	Routing	RIP, OSPF, BGP	
4	**Transport Layer**	UDP, TCP	**12**
5	**Application Layer**	WWW, HTTP, DNS, SMTP, POP3, MIME, IMAP,DHCP, TELNET, SSH, FTP	**12**

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	TCP/IP Protocol Suite	Behrouz A. Forouzan	Tata McGraw Hill	Fourth Edition	2010
2.	Data Communication and Networking	Behrouz A. Forouzan	Tata McGraw Hill	-----	-----
3.	Computer Networks	Andrew Tanenbaum	Pearson	Fifth	2013

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	Marks
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	Unit	Marks
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Computer Networks Practical**
Course Code (refer to student handbook)	**USIT3P3**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Computer Networks Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Configure and manage network devices.
2. Implement and analyze network protocols.
3. Design and simulate network topologies.
4. Ensure network security and troubleshoot issues.
5. Collaborate on network projects and apply knowledge in real-world scenarios.

Curriculum:

Unit	**Title**	**Learning Points**	**No of Lectures**
1	**Practical 1**	Colour code for crimping LAN (Cat5/6/7) cable a. Study of Different color codes b. Study of different connecting devices and their differences c. Crimping LAN Cable	**3**
2	**Practical 2**	Configuring LAN setup a. Planning and Setting IP networks b. Configuring subnet c. Study of basic network command and Network configuration commands. ipconfig, netstat, ARP, ping, trace route etc. d. Basic network troubleshooting. e. Configuration of TCP/IP Protocols in Windows / Linux. f. Implementation of Drive/file sharing and printer sharing.	**3**
3	**Practical 3**	IPv4 Addressing and Subnetting a. Given an IP address and network mask, determine other information about the IP address such as: ● Network address ● Network broadcast address ● Total number of host bits ● Number of hosts b. Given an IP address and network mask, determine other information about the IP address such as:	**3**

		● The subnet address of this subnet ● The broadcast address of this subnet ● The range of host addresses for this subnet ● The maximum number of subnets for this subnet mask ● The number of hosts for each subnet ● The number of subnet bits ● The number of this subnet	
4	**Practical 4**	Designing and configuring a network topology a. Configure IP static routing	**3**
5	**Practical 5**	Configure IP routing using RIP.	**3**
6	**Practical 6**	Configuring Simple and multi-area OSPF.	**3**
7	**Practical 7**	Configuring server and client. a. Configure DHCP b. Configure DNS c. Configure HTTP d. Configure Telnet e. Configure FTP	**3**
8	**Practical 8**	Configure basic security features for networks	**3**
9	**Practical 9**	Packet capture and header analysis by wire-shark (TCP, UDP, IP etc.)	**3**
10	**Practical 10**	Planning and Design a corporate network for a given scenario.	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1	TCP/IP Protocol Suite	Behrouz A. Forouzan	Tata McGraw Hill	Fourth Edition	2010
2	Data Communication and Networking	Behrouz A. Forouzan	Tata McGraw Hill	-----	-----
3	https://www.cisco.com				
4	https://www.computernetworkingnotes.com				

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	**Marks**
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Operating Systems**
Course Code (refer to student handbook)	**USIT304**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Operating Systems

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Role of Operating System Computer System.
2. Use the different types of Operating System and their services.
3. Configure process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
4. Apply virtual memory concepts.
5. Effectively use and manage secondary memory.

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**Operating System Overview**	Objectives and Functions, Evolution, Achievements, Modern Operating Systems, Fault tolerance, OS design considerations for multiprocessor and multicore, overview of different operating systems	**12**
	Processes	Process Description and Control	
II	**Threads, Concurrency**	Mutual Exclusion and Synchronization	**12**
III	**Concurrency**	Deadlock and Starvation	**12**
	Memory	Memory Management, Virtual Memory	
IV	**Scheduling**	Uniprocessor Scheduling, Multiprocessor and Real-Time Scheduling	**12**
V	**IO and File Management**	I/O Management and Disk Scheduling, File Management, Operating System Security	**12**

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Operating Systems – Internals and Design Principles	Willaim Stallings	Pearson	9th	2009

2.	Operating System Concepts	Abraham Silberschatz, Peter B. Galvineg Gagne	Wiley	8th	
3.	Operating Systems	Godbole and Kahate	McGraw Hill	3rd	

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Operating Systems Practical**
Course Code (refer to student handbook)	**USIT3P4**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Operating Systems Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Install and configure operating systems.
2. Manage processes, threads, and system resources.
3. Implement memory management techniques.
4. Handle file system operations and data storage.
5. Troubleshoot and diagnose operating system issues.

Curriculum:

Sr. No.	Title	Learning Points	No of Lectures
1	**Practical 1**	Installation and Configuration of virtual machine a. Installation of virtual machine software. b. Installation of Windows OS c. Installation of Linux OS	**3**
2	**Practical 2**	Windows (DOS) Commands a. Date, time, prompt, md, cd, rd, path. b. Chkdsk, copy, xcopy, format, fidsk, cls, defrag, del, move. c. Diskcomp, diskcopy, diskpart, doskey, echo d. Edit, fc, find, rename, set, type, ver	**3**
3	**Practical 3**	Linux commands: a. pwd, cd, absolute and relative paths, ls, mkdir, rmdir b. file, touch, rm, cp. mv, rename, head, tail, cat, tac, more, less, strings, chmod c. ps, top, kill, pkill, bg, fg d. grep, locate, find, locate e. date, cal, uptime, w, whoami, finger, uname, man, df, du, free, whereis, which f. Compression: tar, gzip	**3**
4	**Practical 4**	Working with Linux Desktop and utilities a. The vi editor b. Graphics User Interface c. Working with Terminal d. Adjusting display resolution e. Using the browsers f. Configuring simple networking g. Creating users and shares	**3**
5	**Practical 5**	Installing utility software on Linux and Windows	**3**

6	**Practical 6**	Running C/C++/Python programs in Linux	**3**
7	**Practical 7**	Introduction to Linux Shell Scripting a. Basic operators b. Decision Making c. Looping d. Regular Expression e. Special variables and command Line arguments	**3**
8	**Practical 8**	Case study of Server OS Windows Server 2022 operating System - Architecture, Components, Services, Configuration	**3**
9	**Practical 9**	Case study of Android OS Architecture, Components, Services, Configuration	**3**
10	**Practical 10**	Case study of Cloud OS AWS, Azure, Google Cloud	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

1. https://ubuntu.com/download/desktop
2. https://sritsense.weebly.com/uploads/5/7/2/7/57272303/android_operating_system.pdf

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	**Marks**
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	Unit	Marks
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Applied Mathematics**
Course Code (refer to student handbook)	**USIT305**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Applied Mathematics

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Apply Knowledge of matrices to solve the problems.
2. Interpret the mathematical results in practical terms of complex numbers.
3. Solve and analyze the differential equations and its applications in related field of computers.
4. Understand various techniques of Laplace transform.
5. Solve multiple integrals.

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**Matrices**	Inverse of a matrix, Properties of matrices, Elementary Transformation, Rank of Matrix, Echelon or Normal Matrix, Linear equations, Linear dependence and linear independence of vectors, Linear transformation, Characteristics roots and characteristics vectors, Properties of characteristic vectors, Caley-Hamilton Theorem, Similarity of matrices, Reduction of matrix to a diagonal matrix which has elements as characteristics values.	**12**
	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, Differentiation and Integration, Graphs of the hyperbolic functions, Logarithms of complex quality, j(=i)as an operator(Electrical circuits)	
II	**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.	**12**

	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, Methods of Substitution	
	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 expiration for the particular integral 1/f(D) X; the general methods, Particular integral : Short methods, Particular integral : Other methods, Differential equations reducible to the linear differential equations with constant coefficients.	
III	**The Laplace Transform**	Introduction, Definition of the Laplace Transform, Table of Elementary Laplace Transforms, Theorems on Important Properties of Laplace Transformation, First Shifting Theorem, Second Shifting Theorem, The Convolution Theorem, Laplace Transform of an Integral and Derivatives	**12**
	Inverse Laplace Transform	Shifting Theorem, Partial fraction Methods, Use of Convolution Theorem, Solution of Ordinary Linear Differential Equations with Constant Coefficients, Solution of Simultaneous Ordinary Differential Equations, Laplace Transformation of Special Function, Periodic Functions, Heaviside Unit Step Function, Dirac-delta Function(Unit Impulse Function)	
IV	**Multiple Integrals**	Double Integral, Change of the order of the integration, Double integral in polar coordinates, Triple integrals.	**12**
	Applications of integration	Areas, Volumes of solids	
V	**Beta and Gamma Functions**	Definitions, Properties and Problems, Duplication formula Differentiation Under the Integral Sign Error Functions	**12**

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	A text book of Applied Mathematics Vol I	P. N. Wartikar and J. N. Wartikar	Pune Vidyarthi Griha		
2.	Applied Mathematics II	P. N. Wartikar and J. N. Wartikar	Pune Vidyarthi Griha		
3.	Higher Engineering Mathematics	Dr. B. S. Grewal	Khanna Publications		

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	Marks
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	Unit	Marks
1	I	12
2	II	12
3	III	12
4	IV	12
5	V	12

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Mobile Programming Practical**
Course Code (refer to student handbook)	**USIT3P5**
Class	**S.Y.B.Sc.IT**
Semester	**III**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Mobile Programming Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Develop mobile applications for various platforms.
2. Utilize mobile development frameworks and tools effectively.
3. Implement user interfaces and user experience (UI/UX) design principles.
4. Integrate mobile applications with backend services and databases.
5. Test, debug, and deploy mobile applications on real devices.

Curriculum:

The practicals will be based on HTML5, CSS, Flutter

Sr. No.	**Title**	**Learning Points**	**No of Lectures**
1	**Practical 0**	Setting up Flutter, PhoneGAP Project and environment.	**3**
2	**Practical 1**	Program to demonstrate the features of Dart language.	**3**
3	**Practical 2**	Designing the mobile app to implement different widgets.	**3**
4	**Practical 3**	Designing the mobile app to implement different Layouts.	**3**
5	**Practical 4**	Designing the mobile app to implement Gestures.	**3**
6	**Practical 5**	Designing the mobile app to implement the theming and styling.	**3**
7	**Practical 6**	Designing the mobile app to implement the routing.	**3**
8	**Practical 7**	Designing the mobile app to implement the animation.	**3**
9	**Practical 8**	Designing the mobile app to implement the state management.	**3**
10	**Practical 9**	Designing the mobile app working with SQLite Database.	**3**
11	**Practical 10**	Designing the mobile app working with Firebase.	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Flutter for Beginners	Alessandro Biessek	Packt Publishing		2019
2.	PhoneGap By Example	Andrey Kovalenko	Packt Publishing	1st	2015

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	**Marks**
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

SEMESTER IV

Name of the Course	**Core Java**
Course Code (refer to student handbook)	**USIT401**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Core Java

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Learn the architecture of Java
2. Identify data types, control flow, classes, inheritance, exceptions and event handling
3. Use object-oriented concepts for problem solving real-life applications
4. Build GUI programs
5. Create event driven programs using java.

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**Introduction**	History, Features of Java, Java Development Kit, Java Application Programming Interface, Java Virtual Machine, Java Program Structure.	**12**
	Classes	The Class Object and Its Attributes, Class Methods, Accessing A Method, Method Overloading, Instantiating Objects from A Class, Constructors, this keyword, super keyword, Types of Classes, Scope Rules, Access Modifier, constants, static members of a class, garbage collection.	
II	**Inheritance**	Derived Class Objects, Inheritance and Access Control, Default Base Class Constructors, this and super keywords. Abstract Classes and Interfaces, Abstract Classes, Abstract Methods,	
	Interfaces	What Is an Interface? How Is an Interface Different from An Abstract Class? Multiple Inheritance, Defining an Interface, Implementing Interfaces.	
III	**Exceptions**	Catching Java Exceptions, Catching Run-Time Exceptions, Handling Multiple Exceptions, The finally Clause, The throws Clause, Built-in Exceptions in java	**12**
	Multithreading	Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, wait() notify() notify all() methods	

	Packages	Introduction to predefined packages, User Defined Packages, Access specifier, Java Built-in packages, Array Class, String Class	
IV	**Introduction to JFC and Swing**	Features of the Java Foundation Classes, Swing API Components, JComponent Class, Containers and Panels, Labels, Buttons, RadioButton, Check Boxes, Text-Entry Components, Menus	**12**
	Layouts	Flow Layout, Grid Layout, Border Layout	
	Event Handling	Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes.	
V	**Advanced Swing Controls**	JScrollPane, Lists and Combo Boxes, Colors and File Choosers, Tables and Trees, JTabbedPane.	**12**
	JDBC	Introduction, JDBC Architecture, JDBC Drivers, java.sql package, Using Statement, PreparedStatement, CallableStatement, ResultSet	

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Core Java 8 for Beginners	Vaishali Shah, Sharnam Shah	SPD	1st	2015
2.	Java: The Complete Reference	Herbert Schildt	McGraw Hill	9th	2014
3.	Murach’s beginning Java with Net Beans	Joel Murach , Michael Urban	SPD	1st	2016
4.	Core Java, Volume I: Fundamentals	Hortsman	Pearson	9th	2013
5.	Core Java, Volume II: Advanced Features	Gary Cornell and Hortsman	Pearson	8th	2008
6.	Core Java: An Integrated Approach	R. Nageswara Rao	DreamTech	1st	2008

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Core Java Practical**
Course Code (refer to student handbook)	**USIT4P1**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Core Java Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Implement Java programs to solve computational problems.
2. Demonstrate proficiency in object-oriented programming (OOP) concepts.
3. Utilize Java libraries and APIs for various functionalities.
4. Develop graphical user interfaces (GUI) using Java Swing or JavaFX.
5. Debug and troubleshoot Java code effectively.

Curriculum:

Sr. No.	Title	Learning Points	No of Lectures
1	**Practical 1**	OOPs concepts in Java – 1 a. Write a program to create a class and implement a default, overloaded and copy Constructor. b. Write a program to create a class and implement the concepts of Method Overloading c. Write a program to create a class and implement the concepts of Static methods	**3**
2	**Practical 2**	OOPs concepts in Java – 2 a. Write a program to implement the concepts of Inheritance and Method overriding b. Write a program to implement the concepts of Abstract classes and methods c. Write a program to implement the concept of interfaces	**3**
4	**Practical 4**	Multithreading: Write a java application to demonstrate 5 bouncing balls of different colors using threads.	**3**
5	**Practical 5**	JDBC a. Write a JDBC program that displays the data of a given table in a GUI Table. b. Write a JDBC program to Show the details of a specified product	**3**

		from a given table selected using Combobox. c. Write a GUI application to Navigate forward and reverse result set data.	
6	**Practical 6**	Swing a. Create a swing application that randomly changes color on button click. b. Create a Swing application to demonstrate use of TextArea using scrollpane to show contest of text file in textarea selected using file chooser. c. Create a Swing application to demonstrate use of scrollpane to change its color selected using colour chooser.	**3**
7	**Practical 7**	Layouts: Write programs for the following layouts: a. Flow Layout b. Grid Layout c. Border Layout	**3**
8	**Practical 8**	Events: Write programs to demonstrate the following events: a. ActionEvent b. MouseEvent c. KeyEvent d. SelectionEvent e. FocusEvent	**3**
9	**Practical 9**	Demonstrate the use of Adapter Class in Event Handling.	**3**
10	**Practical 10**	Demonstrate the use of Anonymous Inner Class in Event Handling	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Core Java 8 for Beginners	Vaishali Shah, Sharnam Shah	SPD	1st	2015
2	Java: The Complete Reference	Herbert Schildt	McGraw Hill	9th	2014
3	Murach’s beginning Java with Net Beans	Joel Murach , Michael Urban	SPD	1st	2016
4	Core Java, Volume I: Fundamentals	Hortsman	Pearson	9th	2013
5	Core Java, Volume II: Advanced Features	Gary Cornell and Hortsman	Pearson	8th	2008
6	Core Java: An Integrated Approach	R. Nageswara Rao	DreamTech	1st	2008

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	Marks
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	Unit	Marks
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Introduction to Embedded Systems**
Course Code (refer to student handbook)	**USIT402**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Introduction to Embedded Systems

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Differentiate between general purpose and embedded systems
2. Discuss the characteristics and quality attributes of embedded systems.
3. Use different types of sensors appropriately.
4. Design and develop embedded systems

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**PIC MICROCONTROLLER**	PIC MICROCONTROLLER: Architecture – memory organization – addressing modes – instruction set – PIC programming in Assembly & C –I/O port, Data Conversion, RAM & ROM Allocation, Timer programming	**12**
	Advanced ARM Controllers	Advanced ARM Controllers: Introduction to ARM and its Features, Architecture – memory organization – addressing modes – The ARM Programmer’s model -Registers – Pipeline - Interrupts – Coprocessors – Interrupt Structure	
II	**Communication Protocol & Implementation**	Communication Protocol & Implementation: Introduction to Communication Protocol, I2C - Interfacing with micro controller using bit-banking method, I2C devices – RTC, Memory, ADC-DAC, Port Expander, SPI (Serial Peripheral Interface), Bluetooth, Wi-Fi and RFID. Understanding Serial, Communication, Bluetooth Communication, SPI Interface ZigBee, Wi-Fi, I2C, Infrared, RFID, GSM, GPS, PDH/SDH/Ethernet	**12**
III	**Getting Started with Arduino**	Getting Started with Arduino: Introduction, Arduino Variants, Install the Drivers, Arduino IDE	**12**
	Basic Functions	Basic Functions: Overview, Structure, Digital I/O Functions, Analog	

		I/O Functions, Advanced I/O Functions, Timer Functions, Communication Functions, Interrupt Functions, Math Functions, Programming Language Reference	
IV	**Using Sensors with the Arduino**	Using Sensors with the Arduino: Light Sensitive Sensors,Temperature Sensors, Temperature and Humidity Sensor, Line-Tracking Sensor, Ultrasonic Sensors, Digital Infrared Motion Sensor, Joystick Module, Gas Sensor, Hall Sensor, Color Sensor, Digital Tilt Sensor, Triple Axis Acceleration Sensor, Analog Sound Sensor, Voice Recognition Module, Digital Vibration Sensor, Flame Sensor, Capacitive Touch Sensor	**12**
	Electromechanical Control Using the Arduino	Electromechanical Control Using the Arduino: DC Motor, Stepper Motor, Servo Motor	
V	**Wireless Control Using the Arduino**	Wireless Control Using the Arduino: Infrared Transmitter and Receiver, Wireless Radio Frequency, Bluetooth, GSM/GPRS, Wi-Fi	**12**
	Case Studies	Case Studies: • Air Quality Monitor Using Arduino • A Fire-Fighting Robot Using Arduino • Intelligent Lock System Using Arduino	

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1	Programming Embedded Systems in C and C++	Michael Barr	O'Reilly	First	1999
2	Introduction to embedded systems	Shibu K V	Tata Mcgraw-Hill	First	2012
3	The 8051 Microcontroller and Embedded Systems	Muhammad Ali Mazidi	Pearson	Second	2011
4	Embedded Systems	Rajkamal	Tata Mcgraw-Hill		

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Introduction to Embedded Systems Practical**
Course Code (refer to student handbook)	**USIT4P2**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Introduction to Embedded Systems Practical

Course Outcomes:

On the successful completion of this course, the learner will be able to

1. Design and implement basic embedded systems using microcontrollers.
2. Interface and control external devices and sensors with embedded systems.
3. Develop real-time embedded software for specific applications.
4. Understand and analyze the interaction between hardware and software in embedded systems.
5. Troubleshoot and debug embedded systems for optimal performance.

Curriculum:

Sr. No.	Title	Learning Points	No of Lectures
1	**Practical 1**	**Introduction to Arduino Introduction to Arduino circuits and breadboarding Blinking of LEDs**	**3**
2	**Practical 2**	**Program using Light Sensitive Sensors**	**3**
3	**Practical 3**	**Program using temperature sensors**	**3**
4	**Practical 4**	**Programs using humidity sensors**	**3**
5	**Practical 5**	**Programs using Line tracking sensors**	**3**
6	**Practical 6**	**Programs using Ultrasonic Sensors**	**3**
7	**Practical 7**	**Programs using digital infrared motion sensors**	**3**
8	**Practical 8**	**Programs using gas sensors**	**3**
9	**Practical 9**	**Programs using servo motors**	**3**
10	**Practical 10**	**Programs making Joystick with Arduino**	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Programming Embedded Systems in C and C++	Michael Barr	O'Reilly	First	1999
2	Introduction to embedded systems	Shibu K V	Tata Mcgraw-Hill	First	2012

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	Marks
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	Unit	Marks
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Computer Oriented Statistical Techniques**
Course Code (refer to student handbook)	**USIT403**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Computer Oriented Statistical Techniques

Course Outcomes:

On the successful completion of the course, students will be able:

1. To calculate and apply measures of central tendencies and measures of dispersion --grouped and ungrouped data cases
2. To calculate the moments, skewness and kurtosis by various methods.
3. To apply discrete and continuous probability distributions to various business problems.
4. To perform Test of Hypothesis as well as calculate confidence interval for a population parameter for single sample and two sample cases. Understand the concept of p-values
5. To apply simple linear regression and correlation model to real life examples.

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**Measures of Central Tendency**	Index, or Subscript, Notation, Summation Notation, Averages, or Measures of Central Tendency ,The Arithmetic Mean, The Weighted Arithmetic Mean ,Properties of the Arithmetic Mean, The Arithmetic Mean Computed from Grouped Data ,The Median ,The Mode, The Empirical Relation Between the Mean, Median, and Mode,The Geometric Mean G, The Harmonic Mean H ,The Relation Between the Arithmetic, Geometric, and Harmonic Means, The Root Mean Square, Quartiles, Deciles, and Percentiles, Software and Measures of Central Tendency.	**12**
	Measures of Dispersion	Dispersion, or Variation, The Range, The Mean Deviation, The Semi-Interquartile Range, The 10–90 Percentile Range, The Standard Deviation, The Variance, Short Methods for Computing the Standard Deviation, Properties of the Standard Deviation, Charlie’s Check, Sheppard’s Correction for Variance, Empirical Relations Between Measures of Dispersion, Absolute and Relative Dispersion; Coefficient of Variation, Standardized Variable; Standard Scores, Software and Measures of Dispersion.	
	Introduction to R	Basic syntax, data types, variables, operators, control statements, R-functions, R –Vectors, R –lists, R Arrays.	

II	**Moments, Skewness, and Kurtosis**	Moments , Moments for Grouped Data, Relations Between Moments, Computation of Moments for Grouped Data, Charlie's Check and Sheppard's Corrections, Moments in Dimensionless Form, Skewness, Kurtosis, Population Moments, Skewness and Kurtosis, Software Computation of Skewness and Kurtosis.	**12**
	Elementary Probability Theory	Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Probability Distributions, Mathematical Expectation, Relation Between Population, Sample Mean, and Variance, Combinatorial Analysis, Combinations, Stirling's Approximation to n!,Relation of Probability to Point Set Theory, Euler or Venn Diagrams and Probability.	
	Elementary Sampling Theory	Sampling Theory, Random Samples and Random Numbers, Sampling With and Without Replacement, Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Differences and Sums, Standard Errors, Software Demonstration of Elementary Sampling Theory.	
III	**Statistical Estimation Theory**	Estimation of Parameters, Unbiased Estimates, Efficient Estimates, Point Estimates and Interval Estimates; Their Reliability, Confidence-Interval Estimates of Population Parameters, Probable Error.	**12**
	Statistical Decision Theory	Statistical Decisions, Statistical Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, Tests Involving Normal Distributions, Two-Tailed and One-Tailed Tests, Special Tests, Operating-Characteristic Curves; the Power of a Test, p-Values for Hypotheses Tests, Control Charts, Tests Involving Sample Differences, Tests Involving Binomial Distributions.	
	Statistics in R	mean, median, mode, Normal Distribution , Binomial Distribution, Frequency Distribution in R.	

IV	**Small Sampling Theory**	Small Samples, Student’s t Distribution, Confidence Intervals, Tests of Hypotheses and Significance, The Chi-Square Distribution, Confidence Intervals for Sigma , Degrees of Freedom, The F Distribution.	**12**
	The Chi-Square Test	Observed and Theoretical Frequencies, Definition of chi-square, Significance Tests, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates’ Correction for Continuity, Simple Formulas for Computing chi-square, Coefficient of Contingency, Correlation of Attributes, Additive Property of chi-square.	
V	**Curve Fitting and the Method of Least Squares**	Relationship Between Variables, Curve Fitting, Equations of Approximating Curves, Freehand Method of Curve Fitting, The Straight Line, The Method of Least Squares, The Least-Squares Line, Nonlinear Relationships, The Least-Squares Parabola, Regression, Applications to Time Series, Problems Involving More Than Two Variables.	**12**
	Correlation Theory	Correlation and Regression, Linear Correlation, Measures of Correlation, The Least-Squares Regression Lines, Standard Error of Estimate, Explained and Unexplained Variation, Coefficient of Correlation, Remarks Concerning the Correlation Coefficient, Product-Moment Formula for the Linear Correlation Coefficient, Short Computational Formulas, Regression Lines and the Linear Correlation Coefficient, Correlation of Time Series, Correlation of Attributes, Sampling Theory of Correlation, Sampling Theory of Regression.	

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Statistics	Murray R. Spiegel, Larry J. Stephens.	McGraw – Hill International	4th	

2.	A Practical Approach using R	R.B. Patil, H.J. Dand and R. Bhavsar	SPD	1st	2017
3.	Fundamental of Mathematical Statistics	S.C. Gupta and V. K. Kapoor	Sultan Chand and Sons	11th Revised	2011
4.	Mathematical Statistics	J. N. Kapur and H.C. saxena	S. Chand	20th Revised	2005

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Computer Oriented Statistical Techniques Practical**
Course Code (refer to student handbook)	**USIT4P3**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Computer Oriented Statistical Techniques Practical

Course Outcomes:

On the successful completion of the course, students will be able to:

1. Apply statistical methods to analyze and interpret data sets.
2. Use statistical software to perform data analysis efficiently.
3. Understand and implement various statistical techniques for hypothesis testing and regression analysis.
4. Present statistical findings visually through graphs and charts.
5. Apply statistical techniques to solve real-world problems in different domains.

Curriculum:

Sr. No.	Title	Learning Points	No of Lectures
1	**Practical 1**	Using R/Python execute the basic commands, array, list and frames.	**3**
2	**Practical 2**	Create a Matrix using R/Python and Perform the operations addition, inverse, transpose and multiplication operations.	**3**
3	**Practical 3**	Using R/Python Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram	**3**
4	**Practical 4**	Using R/Python import the data from Excel / .CSV file and Perform the above functions.	**3**
5	**Practical 5**	Using R/Python import the data from Excel / .CSV file and Calculate the standard deviation, variance, co-variance.	**3**
6	**Practical 6**	Using R/Python import the data from Excel / .CSV file and draw the skewness.	**3**
7	**Practical 7**	Import the data from Excel / .CSV and perform the hypothesis testing.	**3**
8	**Practical 8**	Import the data from Excel / .CSV and perform the Chi-squared Test.	**3**
9	**Practical 9**	Using R/Python perform the binomial and normal distribution on the data.	**3**

10	**Practical 10**	Perform the Linear Regression using R/Python. Compute the Least squares means using R/Python. Compute the Linear Least Square Regression using R/Python	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	A Practical Approach using R	R.B. Patil, H.J. Dand and R. Bhavsar	SPD	1st	2017

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	**Marks**
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Software Engineering**
Course Code (refer to student handbook)	**USIT404**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Software Engineering

Course Outcomes:

On the successful completion of the course, students will be able to:

1. Understand software engineering
2. Apply software engineering principles
3. Discuss various approaches to verification and validation of software including testing, measurements and estimation of software products
4. Create software using different software development models

Curriculum:

Unit	Title	Learning Points	No of Lectures
I	**Introduction**	Introduction: What is software engineering? Software Development Life Cycle, Requirements Analysis, Software Design, Coding, Testing, Maintenance etc.	**12**
	Software Requirements	Software Requirements: Functional and Non-functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements.	
	Software Processes	Software Processes: Process and Project, Component Software Processes.	
	Software Development Process Models	Software Development Process Models. • Waterfall Model. • Prototyping. • Iterative Development. • Rational Unified Process. • The RAD Model • Time boxing Model.	
	Agile software development	Agile software development: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods.	
II	**Socio-technical system**	Socio-technical system: Essential characteristics of socio technical	**12**

		systems, Emergent System Properties, Systems Engineering, Components of system such as organization, people and computers, Dealing Legacy Systems.	
	Critical system	Critical system: Types of critical system, A simple safety critical system, Dependability of a system, Availability and Reliability, Safety and Security of Software systems.	
	Requirements Engineering Processes	Requirements Engineering Processes: Feasibility study, Requirements elicitation and analysis, Requirements Validations, Requirements Management.	
	System Models	System Models: Models and its types, Context Models, Behavioural Models, Data Models, Object Models, Structured Methods.	
III	**Architectural Design**	Architectural Design: Architectural Design Decisions, System Organisation, Modular Decomposition Styles, Control Styles, Reference Architectures.	**12**
	User Interface Design	User Interface Design: Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation.	
	Project Management	Project Management Software Project Management, Management activities, Project Planning, Project Scheduling, Risk Management.	
	Quality Management	Quality Management: Process and Product Quality, Quality assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics.	
IV	**Verification and Validation**	Verification and Validation: Planning Verification and Validation,Software Inspections, Automated Static Analysis, Verification and Formal Methods.	**12**
	Software Testing	Software Testing: System Testing, Component Testing, Test Case Design, Test Automation.	

	Software Measurement	Software Measurement: Size-Oriented Metrics, Function-Oriented Metrics, Extended Function Point Metrics	
	Software Cost Estimation	Software Cost Estimation: Software Productivity, Estimation Techniques, Algorithmic Cost Modelling, Project Duration and Staffing	
V	**Process Improvement**	Process and product quality, Process Classification, Process Measurement, Process Analysis and Modeling, Process Change, The CMMI Process Improvement Framework.	**12**
	Service Oriented Software Engineering	Service Oriented Software Engineering: Services as reusable components, Service Engineering, Software Development with Services.	
	Software reuse	Software reuse: The reuse landscape, Application frameworks,Software product lines, COTS product reuse.	
	Distributed software engineering	Distributed software engineering: Distributed systems issues, Client–server computing, Architectural patterns for distributed systems,Software as a service	

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1	Software Engineering,	Ian Somerville	Pearson Education.	Ninth	
2	Software Engineering	Pankaj Jalote	Narosa Publication		
3	Software engineering, a practitioner’s approach	Roger Pressman	Tata Mcgraw-hill	Seventh	
4	Software Engineering principles and practice	WS Jawadekar	Tata Mcgraw-hill		

5	Software Engineering- A Concise Study	S.A Kelkar	PHI India.		
6	Software Engineering Concept and Applications	Subhajit Datta	Oxford Higher Education		
7	Software Design	D.Budgen	Pearson education	2nd	
8	Software Engineering	KL James	PHI	EEE	2009

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions, single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Software Engineering Practical**
Course Code (refer to student handbook)	**USIT4P4**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Software Engineering Practical

Course Outcomes:

On the successful completion of the course, students will be able to:

1. Apply software development methodologies to manage and execute projects effectively.
2. Design and implement software systems following industry best practices and standards.
3. Collaborate in teams to develop and maintain software products.
4. Utilize software testing techniques to ensure the quality and reliability of software.

Curriculum:

Sr. No.	Title	Learning Points	No of Lectures
1	**Practical 1**	**Study and implementation of class diagrams.**	**3**
2	**Practical 2**	**Study and implementation of Use Case Diagrams.**	**3**
3	**Practical 3**	**Study and implementation of Entity Relationship Diagrams.**	**3**
4	**Practical 4**	**Study and implementation of Sequence Diagrams.**	**3**
5	**Practical 5**	**Study and implementation of State Transition Diagrams.**	**3**
6	**Practical 6**	**Study and implementation of Data Flow Diagrams.**	**3**
7	**Practical 7**	**Study and implementation of Collaboration Diagrams.**	**3**
8	**Practical 8**	**Study and implementation of Activity Diagrams.**	**3**
9	**Practical 9**	**Study and implementation of Component Diagrams.**	**3**
10	**Practical 10**	**Study and implementation of Deployment Diagrams.**	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Object - Oriented Modeling and Design	Michael Blaha, James Rumbaugh	Pearson		2011
2	Learning UML 2. 0	Kim Hamilton, Russ, Miles	O'Reilly Media		2006
3	The unified modeling language user guide	Grady Booch, James Rumbaugh, Ivar Jacobson	Addison-Wesley		2005
4	UML A Beginners Guide	Jason T. Roff	McGraw Hill Professional		2003

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	Marks
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	Unit	Marks
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**

Name of the Course	**Computer Graphics and Animation**
Course Code (refer to student handbook)	**USIT405**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Theory
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Computer Graphics and Animation

Course Outcomes:

On the successful completion of the course, students will be able to:

1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics
2. Compare various algorithms for scan conversion and filling of basic objects
3. Use of geometric transformations on graphics objects and their application in composite form.
4. Extract scene with different clipping methods and its transformation to graphics display device.
5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
6. Render projected objects to naturalize the scene in 2D view and use of illumination models
7. Understand the core concepts and mathematical foundations of computer graphics
8. Know the fundamental computer graphics algorithms and data structures
9. Understand an overview of different modeling approaches and methods
10. Apply basic shading and texture mapping techniques
11. Understand light interaction with 3D scenes
12. Explain the applications, areas, and graphic pipeline, display and hardcopy technologies.
13. Apply and compare the algorithms for drawing 2D images also explain aliasing, anti-aliasing and half toning techniques.
14. Discuss OpenGL application programming Interface and apply it for 2D & 3D computer graphics.
15. Analyze and apply clipping algorithms and transformation on 2D images.
16. Solve the problems on viewing transformations and explain the projection and hidden surface removal algorithms.
17. Apply basic ray tracing algorithm, shading, shadows, curves and surfaces and also solve the problems of curves.

Curriculum:

Unit	**Title**	**Learning Points**	**No of Lectures**
I	**Introduction to Computer Graphics**	Introduction to Computer Graphics: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for	**12**

		Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays.	
	Scan conversion	Scan conversion – Digital Differential Analyzer (DDA) algorithm, Bresenhams' Line drawing algorithm. Bresenhams' method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Clipping Lines algorithms–Cyrus-Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.	
II	**Two-Dimensional Transformations**	Two-Dimensional Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations.	**12**
	Three-Dimensional Transformations	Three-Dimensional Transformations: Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-	

		Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.	
III	**Viewing in 3D**	Viewing in 3D Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, The Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid.	**12**
	Light	Light: Radiometry, Transport, Equation, Photometry	
	Color	Color: Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance	
IV	**Visible-Surface Determination**	Visible-Surface Determination: Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter’s algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods.	**12**
	Plane Curves and Surfaces	Plane Curves and Surfaces: Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Parametric Representation of an	

		Ellipse, Parametric Representation of a Parabola, Parametric Representation of a Hyperbola, Representation of Space Curves, Cubic Splines, , Bezier Curves, B-spline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces.	
V	**Computer Animation**	Computer Animation: Principles of Animation, Key framing, Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects.	**12**
	Image Manipulation and Storage	Image Manipulation and Storage: What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.	

Learning Resources recommended:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1	Computer Graphics - Principles and Practice	J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes	Pearson	2 nd	
2	Steve Marschner, Peter Shirley	Fundamentals of Computer Graphics	CRC press	4th	2016
3	Computer Graphics	Hearn, Baker	Pearson	2nd	
4	Principles of Interactive Computer Graphics	William M. Newman and Robert F. Sproull	TMH	2nd	

5	Mathematical Elements for CG	D. F. Rogers, J. A. Adams	TMH	2nd	

Evaluation Pattern

A. Continuous Internal Evaluation (40 Marks):

Method	**Marks**
Unit Test (MCQ / Descriptive – Based on Theory and/or Problems Online/Offline – 1 unit test of 20 marks	20
Assignments	10
Attendance and active participation in classroom	10

B. Semester End Evaluation (Paper Pattern) (60 Marks – 2 hours):

Question No	**Unit**	**Marks**
1	**I**	**12**
2	**II**	**12**
3	**III**	**12**
4	**IV**	**12**
5	**V**	**12**

Guidelines for paper pattern for Semester End Evaluation:

1. All questions will be compulsory.
2. Descriptive type of questions, programming-based questions, problem solving / numericals based questions,single line answers,etc., will contain internal options.
3. Refer **appendix I** for Paper pattern.

Name of the Course	**Computer Graphics and Animation Practical**
Course Code (refer to student handbook)	**USIT4P5**
Class	**S.Y.B.Sc.IT**
Semester	**IV**
No of Credits	**2**
Nature	Practical
Type (applicable to NEP only)	Core
Highlight revision specific to employability/ entrepreneurship/ skill development (if any) 100 words	

Nomenclature: Computer Graphics and Animation Practical

Course Outcomes:

On the successful completion of the course, students will be able to:

1. Develop interactive computer graphics applications using graphics libraries and APIs.
2. Implement 2D and 3D transformations, projections, and rendering techniques.
3. Create visually appealing animations and special effects.
4. Understand and apply principles of computer graphics to solve graphical challenges.
5. Collaborate in teams to design and showcase computer graphic projects.

Curriculum:

Sr. No.	**Title**	**Learning Points**	**No of Lectures**
1	**Practical 1**	Solve the following: a. Study and enlist the basic functions used for graphics in C / C++ / Python language. Give an example for each of them. b. Draw a co-ordinate axis at the center of the screen.	**3**
2	**Practical 2**	Solve the following: a. Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each region with appropriate message. b. Draw a simple hut on the screen.	**3**
3	**Practical 3**	Draw the following basic shapes in the center of the screen : i. Circle ii. Rectangle iii. Square iv. Concentric Circles v. Ellipse vi. Line	**3**
4	**Practical 4**	Solve the following: a. Develop the program for DDA Line drawing algorithm. b. Develop the program for Bresenham’s Line drawing algorithm.	**3**
5	**Practical 5**	Solve the following: a. Develop the program for the mid-point circle drawing algorithm. b. Develop the program for the mid-point ellipse drawing algorithm.	**3**
6	**Practical 6**	Solve the following: a. Write a program to implement 2D scaling. b. Write a program to perform 2D translation	**3**
7	**Practical 7**	Solve the following:	**3**

		a. Perform 2D Rotation on a given object. b. Program to create a house like figure and perform the following operations. i. Scaling about the origin followed by translation. ii. Scaling with reference to an arbitrary point. iii. Reflect about the line y=mx+c	
8	**Practical 8**	Solve the following: a. Write a program to implement Cohen-Sutherland clipping. b. Write a program to implement Liang - Barsky Line Clipping Algorithm	**3**
9	**Practical 9**	Solve the following: a. Write a program to fill a circle using Flood Fill Algorithm. b. Write a program to fill a circle using Boundary Fill Algorithm.	**3**
10	**Practical 10**	Solve the following: a. Develop a simple text screen saver using graphics functions. b. Perform smiling face animation using graphic functions. c. Draw the moving car on the screen.	**3**

This is sample Practical list. Course instructor may change the practical as per syllabus.

Learning Resources recommended:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1	Computer Graphics - Principles and Practice	J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes	Pearson Education	Second Edition	
2	Steve Marschner, Peter Shirley	Fundamentals of Computer Graphics	CRC press	Fourth Edition	2016
3	Computer Graphics	Hearn, Baker	Pearson Education	Second	
4	Principles of Interactive Computer Graphics	William M. Newman and Robert F. Sproull	Tata McGraw Hill	Second	

Evaluation Pattern

A. Continuous Internal Evaluation (20 Marks):

Method	Marks
Certified Journal	10
Attendance and active participation in Laboratory	10

B. Semester End Evaluation (Paper Pattern) (30 Marks – 2 hours):

Question No	Unit	Marks
1	**Program 1**	**10**
2	**Program 2**	**15**
3	**Viva**	**05**