Academic Council
Item No:

UNIVERSITY OF MUMBAI

Syllabus for S.Y.B.Sc.

Programme: B.Sc.

Course: Information Technology

Choice Based Credit and Semester System

with effect from the academic year

2023 – 2024

(This page is intentionally left blank)

UNIVERSITY OF MUMBAI

Syllabus for Approval

Sr. No.	Heading	Particulars
1	Title of the Programme	**Bachelor of Science in Information Technology (Second Year)**
2	Eligibility for Admission	Ordinance no. O.5051 Circular no. UG/284 of 2007 dated 16th June 2007
3	Passing Marks	40%
4	Ordinances / Regulations (if any)	As applicable for all B.Sc. Courses
5	No. of Years / Semesters	Three years – Six Semesters
6	Level	~~P.G.~~ / U.G./ ~~Diploma~~ / ~~Certificate~~ **(Strike out which is not applicable)**
7	Pattern	~~Yearly~~ / Semester **(Strike out which is not applicable)**
8	Status	Revised / ~~New~~ /~~Amended~~ **(Strike out which is not applicable)**
9	To be implemented from Academic Year	From Academic Year **2023-2024**

(This page is intentionally left blank)

Semester – 3			
Course Code	**Course Type**	**Course Title**	**Credits**
USIT301	Skill Enhancement Course	Python Programming	2
USIT302	Core Subject	Data Structures	2
USIT303	Core Subject	Computer Networks	2
USIT304	Core Subject	Operating Systems	2
USIT305	Core Subject	Applied Mathematics	2
USIT3P1	Skill Enhancement Course Practical	Python Programming Practical	2
USIT3P2	Core Subject Practical	Data Structures Practical	2
USIT3P3	Core Subject Practical	Computer Networks Practical	2
USIT3P4	Core Subject Practical	Operating Systems Practical	2
USIT3P5	Core Subject Practical	Mobile Programming Practical	2
		Total Credits	**20**

Semester – 4			
Course Code	**Course Type**	**Course Title**	**Credits**
USIT401	Skill Enhancement Course	Core Java	2
USIT402	Core Subject	Introduction to Embedded Systems	2
USIT403	Core Subject	Computer Oriented Statistical Techniques	2
USIT404	Core Subject	Software Engineering	2
USIT405	Core Subject	Computer Graphics and Animation	2
USIT4P1	Skill Enhancement Course Practical	Core Java Practical	2
USIT4P2	Core Subject Practical	Introduction to Embedded Systems Practical	2
USIT4P3	Core Subject Practical	Computer Oriented Statistical Techniques Practical	2
USIT4P4	Core Subject Practical	Software Engineering Practical	2
USIT4P5	Core Subject Practical	Computer Graphics and Animation Practical	2
		Total Credits	**20**

(This page is intentionally left blank)

Table of Contents

SEMESTER III

(This page is intentionally left blank)

Python Programming

B. Sc. (Information Technology)		Semester – III	
Course Name: Python Programming		**Course Code: USIT301**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

- Interpret the fundamental Python syntax and semantics and be fluent in the use of Python control flow statements.
- Express proficiency in the handling of strings and functions.
- Determine the methods to create and manipulate Python programs by utilizing the data structures like lists, dictionaries, tuples and sets.
- Identify the commonly used operations involving file systems and regular expressions.
- Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism as used in Python.

Unit	Details	Lectures
I	**Introduction:** The Python Programming Language, History, features, Installing Python, Running 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	**12**
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 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.	**12**
III	**Lists:** Values and Accessing Elements, Lists are mutable, traversing a List, Deleting elements from List, Built-in List Operators,	**12**

	Concatenation, Repetition, In Operator, Built-in List functions and methods **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. **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	**12**
V	**Creating the GUI Form and Adding 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. **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.	**12**

Books and References:

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

Course Outcome:

After completing the course, the learner will be able to:

CO1: Aware of the variables, expressions, looping and conditions used in Python programming.

CO2: Implement functions, strings, lists, tuples and directories

CO3: Create GUI forms and add widgets.

CO4: Use MySQL to store data.

CO5: Apply the programming skillset learnt here into various domains by having advance programming skillset of Python and usage of libraries.

Data Structures

B. Sc. (Information Technology)		Semester – III	
Course Name: Data Structures		**Course Code: USIT302**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

- Ability to analyze the performance of algorithms.
- Ability to choose appropriate algorithm design techniques for solving problems.
- Understand how the choice of data structures and the algorithm design methods impact the performance of programs.

Unit	Details	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. **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.	**12**
II	**Linked List:** Linked List, One-way Linked List, Traversal of Linked List, Searching, Memory Allocation and De-allocation, Insertion in Linked List, Deletion from Linked List, Copying a List into Other 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,	**12**

	Evaluation of Arithmetic Expression, Matching Parenthesis, infix and postfix operations, Recursion**.** **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. **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.	**12**
V	**Hashing Techniques** Hash function, Address calculation techniques, Common hashing functions Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and rehashing **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.	**12**

Books and References:

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 McGraw 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

Course Outcome:

After completing the course, the learner will be able to:

CO1: Identify and distinguish data structure classification, data types, their complexities

CO2: Implement array, linked list, stack and queue.

CO3: Implement trees, various hashing techniques and graph for various applications

CO4: Compare various sorting and searching techniques

Computer Networks

B. Sc. (Information Technology)		Semester – III	
Course Name: Computer Networks		**Course Code: USIT303**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

- Knowledge of uses and services of Computer Network.
- Ability to identify types and topologies of network.
- Understanding of analog and digital transmission of data.
- Familiarization with the techniques of routing.
- Understand the functioning of networking application

Unit	Details	Lectures
I	**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**
II	**Introduction to Physical layer:** Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance. **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.	**12**
III	**Network Layer:** IPv4 Addresses, IPv4 Protocol, ARP, ICMP, IPv6 **Routing:** RIP, OSPF, BGP	**12**
IV	**Transport Layer:** UDP, TCP	**12**
V	**Application Layer:** WWW, HTTP, DNS, SMTP, POP3, MIME, IMAP, DHCP, TELNET, SSH, FTP	**12**

Books and References:

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

Online Resources:

- https://ekumbh.aicte-india.org/allbook.php
- https://free.aicte-india.org/

Course Outcomes:

After completing the course, the learner will be able to:

CO1: Identify various data communication standards, topologies and terminologies

CO2: Describe how signals are used to transfer data and communication aspects between nodes

CO3: Configure IP addresses using TCP/IP protocol suite

CO4: Use different application layer protocols

Operating Systems

B. Sc. (Information Technology)		Semester – III	
Course Name: Operating Systems		**Course Code: USIT304**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

- Analyze the concepts of processes in operating system and illustration of the scheduling of processor for a given problem instance.
- Identify the dead lock situation and provide appropriate solution so that protection and security of the operating system is also maintained.
- Analyze memory management techniques, concepts of virtual memory and disk scheduling.
- Understand the implementation of file systems and directories along with the interfacing of IO devices with the operating system.
- Ability to apply CPU scheduling algorithms to manage tasks.
- Initiation into the process of applying memory management methods and allocation policies.
- Knowledge of methods of prevention and recovery from a system deadlock.

Unit	Details		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 **Processes:** Process Description and Control.		**12**
II	**Threads, Concurrency:** Mutual Exclusion and Synchronization.		**12**
III	**Concurrency:** Deadlock and Starvation, **Memory:** Memory Management, Virtual Memory.		**12**
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**

Books and References:

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,	Wiley	8th	

		Peter B. Galvineg Gagne			
3.	Operating Systems	Godbole and Kahate	McGraw Hill	3rd	

Online Resources:

- https://onlinecourses.nptel.ac.in/noc20_cs04/preview
- https://free.aicte-india.org/
- https://www.javatpoint.com/best-courses-for-the-operating-system

Course Outcomes:

After completing the course, the learner will be able to:

CO1: Role of Operating System Computer System.

CO2: Use the different types of Operating System and their services.

CO3: configure process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.

CO4: Apply virtual memory concepts.

CO5: Effectively use and manage secondary memory.

Applied Mathematics

B. Sc. (Information Technology)		Semester – III	
Course Name: Applied Mathematics		**Course Code: USIT305**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

The course is aimed to develop the basic Mathematical skills of IT students that are imperative for effective understanding of IT subjects.

- Apply the knowledge of matrices to solve the problems.
- Know and to understand various types of numerical methods.
- Ability to interpret the mathematical results in physical or practical terms for complex numbers.
- Inculcate the habit of Mathematical Thinking through Indeterminate forms and Taylor series expansion
- Solve and analyze the Partial derivatives and its application in related field of engineering

Unit	Details	Lectures
I	**Matrices:** Inverse of a matrix, Properties of matrices, Elementary Transformation, Rank of Matrix, Echelon or Normal Matrix, Inverse of 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. **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)	**12**
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. **Differential equation of the first order of a degree higher than the first:** Introduction, Solvable for p (or the method of factors), Solve for	**12**

	y, Solve for x, Clairaut's form of the equation, Methods of Substitution, Method 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, Laplace Transform of Derivatives, **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),	**12**
IV	**Multiple Integrals:** Double Integral, Change of the order of the integration, Double integral in polar co-ordinates, Triple integrals. **Applications of integration:** Areas, Volumes of solids.	**12**
V	**Beta and Gamma Functions –** Definitions**,** Properties and Problems. Duplication formula. **Differentiation Under the Integral Sign** **Error Functions**	**12**

Books and References:

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 Vidyathi Graha		
2.	Applied Mathematics II	P. N. Wartikar and J. N. Wartikar	Pune Vidyathi Graha		
3.	Higher Engineering Mathematics	Dr. B. S. Grewal	Khanna Publications		

Course Outcomes:

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

CO 1: Solve the matrix operations, identify the linear dependence and independence of a vectors.

CO 2: Familiar with the various forms and operations of a complex number.
CO 3: Find the Laplace transform of a function and Inverse Laplace transform of a function using definition also solve ordinary differential equations using Laplace transform.
CO 4: Evaluate the multiple integrals in Cartesian, Polar coordinates, change the order of the integral,
CO 5: Apply integration methods to calculate the areas and volumes of solids.
CO 6: Evaluate the Beta, Gamma, Differentiation Under integral sign and error functions

Python Programming Practical

B. Sc. (Information Technology)		Semester – III	
Course Name: Python Programming Practical		**Course Code: USIT3P1**	
Periods per week (1 Period is 50 minutes)		**3**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**
	Internal	**--**	**--**

List of 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.
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.
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.	**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.
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
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 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.
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.
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.
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 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.
b.	Write a program to implement exception handling.
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 Message, Button, Entry, Checkbutton, Radiobutton, Scale etc.
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.

Books and References:

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

Data Structures Practical

B. Sc. (Information Technology)		Semester – III	
Course Name: Data Structures Practical		**Course Code: USIT3P2**	
Periods per week (1 Period is 50 minutes)		**3**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**
	Internal	**--**	**--**

List of Practical	
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]
2.	**Implement the following for Linked List:**
a.	Write a program to create a single 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 double linked list and sort the elements in the linked list.
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 Tower of Hanoi problem.
4.	**Implement the following for Queue:**
a.	Write a program to implement the concept of Queue with Insert, Delete, Display and Exit operations.
b.	Write a program to implement the concept of Circular Queue
c.	Write a program to implement the concept of Deque.
5.	**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.
6.	**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.
7.	**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
8.	**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.
9.	**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.
10.	**Implement the following data structure techniques:**
a.	Write a program to generate the adjacency matrix.
b.	Write a program for shortest path diagram.

Books and References:

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

Computer Network Practical

B. Sc. (Information Technology)		Semester – III	
Course Name: Computer Network Practical		**Course Code: USIT3P3**	
Periods per week (1 Period is 50 minutes)		**3**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**
	Internal	**--**	

List of Practical:	
1.	**Colour code for crimping LAN (Cat 5/6/7) cable**
a.	Study of Different color codes
b.	Study of different connecting devices and their differences
c.	Crimping LAN Cable
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.	**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: • 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.	**Designing and configuring a network topology**
a.	Configure IP static routing

5.	Configure IP routing using RIP.
6.	Configuring Simple and multi-area OSPF.
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.	Packet capture and header analysis by wire-shark (TCP, UDP, IP etc.)
10.	**Planning and Design a corporate network for a given scenario.**

Operating System Practical

B. Sc. (Information Technology)		Semester – III	
Course Name: Operating System Practical		**Course Code: USIT3P4**	
Periods per week (1 Period is 50 minutes)		**3**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**
	Internal	**--**	

List of Practical:	
1.	**Installation and Configuration of virtual machine**
d.	Installation of virtual machine software.
e.	Installation of Windows OS
f.	Installation of Linux OS
2.	**Windows (DOS) Commands**
g.	Date, time, prompt, md, cd, rd, path.
h.	Chkdsk, copy, xcopy, format, fidsk, cls, defrag, del, move.
i.	Diskcomp, diskcopy, diskpart, doskey, echo
j.	Edit, fc, find, rename, set, type, ver
3.	**Linux commands:**
c.	pwd, cd, absolute and relative paths, ls, mkdir, rmdir
d.	file, touch, rm, cp. mv, rename, head, tail, cat, tac, more, less, strings, chmod
e.	ps, top, kill, pkill, bg, fg
f.	grep, locate, find, locate
g.	date, cal, uptime, w, whoami, finger, uname, man, df, du, free, whereis, which
h.	Compression: tar, gzip
4.	**Working with Linux Desktop and utilities**
b.	The vi editor
c.	Graphics User Interface
d.	Working with Terminal
e.	Adjusting display resolution
f.	Using the browsers
g.	Configuring simple networking
h.	Creating users and shares
5.	**Installing utility software on Linux and Windows**
6.	**Running C/C++/Python programs in Linux**
7.	**Introduction to Linux Shell Scripting**
f.	Basic operators

g.	Decision Making
h.	Looping
i.	Regular Expression
j.	Special variables and command Line arguments
8.	**Case study of Server OS:** Windows Server 2022 operating System - Architecture, Components, Services, Configuration
9.	**Case study of Android OS:** Architecture, Components, Services, Configuration
10.	**Case study of Cloud OS:** AWS, Azure, Google Cloud

Mobile Programming Practical

B. Sc. (Information Technology)		Semester – III	
Course Name: Mobile Programming Practical		**Course Code: USIT3P5**	
Periods per week (1 Period is 50 minutes)		**3**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**
	Internal	**--**	**--**

The practical's will be based on HTML5, CSS, Flutter. (Android will be introduced later after they learn Java)

List of Practical	
	Setting up Flutter, PhoneGAP Project and environment.
1.	Program to demonstrate the features of Dart language.
2.	Designing the mobile app to implement different widgets.
3.	Designing the mobile app to implement different Layouts.
4.	Designing the mobile app to implement Gestures.
5.	Designing the mobile app to implement the theming and styling.
6.	Designing the mobile app to implement the routing.
7.	Designing the mobile app to implement the animation.
8.	Designing the mobile app to implement the state management.
9.	Designing the mobile app working with SQLite Database.
10.	Designing the mobile app working with Firebase.

Books and References:					
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

(This page is intentionally left blank)