LOK JAGRUTI UNIVERSITY (LJU)

INSTITUTE OF ENGINEERING & TECHNOLOGY

Department of Information Technology (702)

Bachelor of Technology (B.E.) – Semester – III

Course Code:	**017022391**
Course Name:	**Fundamentals of Computer Science using Python - I**
Category of Course:	Engineering Science Course (ESC)
Prerequisite Course:	Computer Programming using C (017022191)

Teaching Scheme				
Lecture (L)	**Tutorial (T)**	**Practical (P)**	**Credit**	**Total Hours**
2	**0**	**6**	**5**	**20**

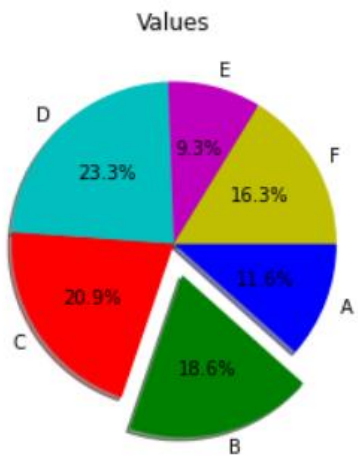
Syllabus

Unit No.	Topic	Prerequisite Topic	Successive Topic	Teaching Hours
01	**Introduction to Python and Jupyter Notebooks**			**1 (5%)**
	1.1 Basic Data Types -integers, Booleans, floats, strings, type() function, typecasting	---	---	
	1.2 Declaring and using variables, Comments, Reading input from users, printing output	---		
	1.3 Arithmetic Operators: +, -, *, /, **, //, % Logical Operators: and, or, not Membership Operators: in, not in Assignment Operators: =, +=, -=, *=, /=, **=, %= Comparison Operators: ==, !=, >, <, >=, <= Ternary Operators, Operator precedence	Classification of Operators - Arithmetic, Relational, Logical, Assignment, Increment/Decrement, Bitwise, Special, Unary, Binary, and Ternary Operators (017022191-Unit-2.4)		
	1.4 Using Interactive Shell, Editing, saving, and running a script	---		
	1.5 Jupyter Notebooks: Creating, Opening, Saving and Downloading Notebooks, running cells	---		
02	**Conditional Execution and Iterations**			**2 (15%)**
	2.1 Control Statements: Simple if, if-else, if-elif-else, Nested if	---	---	
	2.2 Loops: For using range function with else, while loop with else			
	2.3 Break, Continue, Pass Statement			
03	**Functions, Scoping and Abstraction**			**1 (5%)**
	3.1 Declaring, Defining and Invoking a Function, Function Specification	---	---	
	3.2 Function arguments: keyword, default, positional, variable-length			
	3.3 Local v/s Global variables			
04	**Immutable Data Structures**			**2 (10%)**
	4.1 Strings -immutability, declaring, accessing through for loop, slicing, concatenation, len, capitalize(), enumerate(), isalnum(), isalpha(), islower(), isupper(), lower(), upper(), isnumeric(), find(), index(), split(), strip(), removing punctuation using translate(), count()	---	---	
	4.2 Tuples -immutability, create, assign, access, del, slicing, concatenation, comparing tuples using >, <, ==, count() function			
	4.3 Built-in functions -sorted, reversed, min, max			
05	**Mutable Data Structures**			**4 (15%)**
	5.1 Lists -mutability, declaring, accessing through for loop, append(), count(), extend(), index(), insert(), pop(), remove(), reverse(), sort()	---		
	5.2 Dictionaries -create, access using for loop, clear(), copy(), fromkeys(), get(), has_key(), items(), keys(), update(), values()			
	5.3 Sets -declaring, accessing, set(), frozenset(), issubset(), issuperset(), union(), intersection(), difference(), symmetric_difference(), copy()			
	5.4 Lambda functions: Map, Reduce, Filter			
06	**Working with Files**			**1 (10%)**
	6.1 Working with text files, opening files -open in w, r and a modes with open function	---	---	
	6.2 Reading data from files -read(), readline(), readlines(), seek() and tell() functions			
	6.3 Writing data to files -write(), writelines(), closing files			
07	**Modules, Directories**			**1 (5%)**
	7.1 Modules and Packages -difference between import and from…import	---	---	

	7.2 Python OS Module -getcwd(), chdir(), mkdir(), listdir(), remove(), rmdir()			
08	**Introduction to Object Oriented Programming and Exception Handling**			**2 (8%)**
	8.1 Abstraction & Encapsulation: Defining classes, attributes, methods, constructors, destructors, objects, generators	---	---	
	8.2 Exception Handling -try, except, finally, custom exceptions			
09	**Advanced OOP Concepts and Introduction to NumPy**			**2 (17%)**
	9.1 Polymorphism -Overloading and overriding			
	9.2 Inheritance -single, multi-level, Method Resolution Order (MRO), Abstract Class			
	9.3 Arrays -Creating a NumPy ndarray Object, 1D, 2D and 3D Arrays	---	---	
	9.4 Array Indexing, Slicing, Shape, Reshaping, Iteration			
	9.5 Built-in functions -concatenate, array_split, where, sort			
10	**Introduction to Matplotlib**			**4 (10%)**
	10.1 Basic Methods -Plot() with different markers, colors, and linestyles, show() Describing the plot -xlabel, ylabel, title with different fonts, colors and positions Grids and subplots	---	---	
	10.2 Simple Scatterplots with color, size and alpha Simple Bar charts bar() and barh() Simple Histogram with hist() Simple pie charts with pie along with labelling and coloring			

Sr No.	Practical Title	Link to Theory Syllabus
1	Write a Python program to check whether the given no is Armstrong or not using user defined function.	Unit-3
2	Write a python program to implement is Palindrome() function to check given string is palindrome or not.	Unit-3
3	Write a Python program to find reverse of given number using user defined function.	Unit-3
4	Create a Python program to read a text file and do following: 1. Print no. of lines 2. Print no. of unique words 3. Store each word with its occurrence in dictionary.	Unit-4
5	Write a python program that reads a text file containing full-name of the employees of a company and then from these names it extracts Last-name of employee and stores them in sorted order in new file. Input Text file contains names in format: First-name Middle-name Last-name	Unit-4, 6
6	Write a python program to read line by line from a given files file1 & file2 and write into file3.	Unit-4, 6
7	Write a Python program which will return the sum of the numbers in the array, returning 0 for an empty array. Except the number 13 is very unlucky, so it does not count and number that come immediately after 13 also do not count. Example : [1, 2, 3, 4] = 10 [1, 2, 3, 4, 13] = 10 [13, 1, 2, 3, 13] = 5	Unit-5
8	Write a Python program which takes a list and returns a list with the elements "shifted left by one position" so [1, 2, 3] yields [2, 3, 1]. Example: [1, 2, 3] → [2, 3, 1] [11, 12, 13] → [12, 13, 11]	Unit-5
9	Write a Python program to create class GTU with attributes like class variable cnt, instance variables x and y, instance methods get_value and print_value.	Unit-8
10	Create a class Employee with data members: name, department and salary. Create suitable methods for reading and printing employee information	Unit-8
11	Write a Python program to implement the concept of inheritance.	Unit-9
12	Write a python program to implement Student class which has method to calculate CGPA. Assume suitable class variables.	Unit-8
13	Write a python program in which Maruti and Santro sub classes implement the abstract methods of the super class Car.	Unit-9
14	Create a class student with following member attributes: roll no, name, age and total marks. Create suitable methods for reading and printing member variables. Write a python program to overload '==' operator to print the details of students having same marks.	Unit-9
15	Create a bar chart for the following lists: x=['one', 'two', 'three', 'four', 'five'] y=[5, 24, 35, 67, 12]	Unit-10
16	Create a scatterplot for the following data points: x = [1, 2, 3, 4] y = [4, 1, 3, 6]	Unit-10
17	Create a pie chart for the following data: [35, 25, 25, 15] With the following labels: ["Apples", "Bananas", "Cherries", "Dates"]	Unit-10
18	Explain the use of the following functions with code and output: getcwd(), chdir(), mkdir(), listdir(), remove(), rmdir()	Unit-7
19	Demonstrate the use of try, except and finally with code.	Unit-8
20	Take three variables and get their values from the user. Without using any sorting functions or Python built-in data structures (lists, tuples, dictionaries, etc.), sort these numbers in ascending and descending orders.	Unit -2
21	Calculate the number of basic American coins given a value less than 1 dollar. A penny is worth 1 cent, a nickel is worth 5 cents, a dime is worth 10 cents, and a quarter is worth 25 cents. It takes 100 cents to make 1 dollar. So given an amount less than 1 dollar (if using floats, convert to integers for this exercise), calculate the number of each type of coin necessary to achieve the amount, maximizing the number of larger denomination coins. For example, given $0.76, or 76 cents, the correct output would be "3 quarters and 1 penny." Output such as "76 pennies" and "2 quarters, 2 dimes, 1 nickel, and 1 penny" are not acceptable.	Unit -2
22	Write a Python program that prompts the user to enter numbers and stops only when the user enters "done". After this, print the average of the numbers and the minimum number and the maximum number from among all the numbers entered by the user. Note: You are not allowed to use any built-in structures like lists, tuples, etc. or any built-in functions like max, min, etc. E.g., if the user enters: 3, 4, 1, 6 ,"done" The program should print the Average: 3.5 Minimum Number: 1 Maximum Number: 6	Unit -2
23	Write a program to ask the user to input a list of names, in the format "Last Name, First Name," i.e., last name, comma, first name. Write a function that manages the input so that when/if the user types the names in the wrong order, i.e., "First Name Last Name," the error is corrected, and the user is notified. This function should also keep track of the number of input mistakes. When the user is done, sort the list, and display the sorted names in "Last Name, First Name" order.	Unit -4
24	a) Create a function called findchr(), with the following declaration: def findchr(string, char)	Unit -4

	findchr() will look for character char in string and return the index of the first occurrence of char, or -1 if that char is not part of string. You cannot use string.find() or string.index() functions or methods. (b) Create another function called rfindchr() that will find the last occurrence of a character in a string. Naturally this works similarly to findchr(), but it starts its search from the end of the input string. (c) Create a third function called subchr() with the following declaration: def subchr(string, origchar, newchar) subchr() is similar to findchr() except that whenever origchar is found, it is replaced by newchar. The modified string is the return value. Use of any substring or replace built-in string methods is prohibited.	
25	Given an integer value, return a string with the equivalent English text of each digit. For example, an input of 89 results in "eighty-nine" being returned. Restrict values to be between 0 and 1,000.	Unit -3
26	Create a function that will return another string similar to the input string, but with its case inverted. For example, input of "Mr. Ed" will result in "mR. eD" as the output string. Note: Use of the built-in swapcase function is prohibited.	Unit -4
27	Given a list L of size N. You need to count the number of special elements in the given list. An element is special if removal of that element makes the list balanced. The list will be balanced if sum of even index elements is equal to the sum of odd index elements. Example Input Input 1: A = [2, 1, 6, 4] Input 2: A = [5, 5, 2, 5, 8] Example Output Output 1: 1 Output 2: 2 Explanation 1: After deleting 1 from list : [2,6,4] (2+4) = (6) Hence 1 is the only special element, so count is 1 Explanation 2: If we delete A[0] or A[1] , list will be balanced (5+5) = (2+8) So A[0] and A[1] are special elements, so count is 2.	Unit -5
28	You own a pizzeria named Olly's Pizzas and want to create a Python program to handle the customers and revenue. Create the following classes with the following methods: Class Pizza containing 1. init method: to initialize the size (small, medium, large), toppings (corn, tomato, onion, capsicum, mushroom, olives, broccoli), cheese (mozzarella, feta, cheddar). Note: One pizza can have only one size but many toppings and cheese. 2. price method: to calculate the prize of the pizza in the following way: small = 50, medium = 100, large = 200 Each topping costs 20 rupees extra, except broccoli, olives and mushroom, which are exotic and so cost 50 rupees each. Each type of cheese costs an extra 50 rupees. Class Order containing 1. init method: to initialize the name, customerid of the customer who placed the order 2. order method: to allow the customer to select pizzas with choice of toppings and cheese 3. bill method: to generate details about each pizza ordered by the customer and the total cost of the order. Note: A customer can get multiple pizzas in one order. Create appropriate objects of these classes.	Unit -8
29	Create a class called StackQueue that mimics the behaviour of both, a stack and a queue. It should contain the following methods: init() to initialize the StackQueue list holding the values shift() returns the first element and removes it from the list unshift() "pushes" a new element to the front or head of the list push() adds a new element to the end of a list pop() returns the last element and removes it from the list display() returns the contents of the list Generate custom exceptions for both shift and pop methods if there are no elements left in the list to remove. Create an object of this class and show its operation with correct output.	Unit -7, 8
30	Write a "pager" program. Your solution should prompt for a filename, and display the text file 25 lines at a time, pausing each time to ask the user to enter the word "continue", in order to show the next 25 lines or enter the word "stop" to close the file.	Unit -6

31	Define a class called Fraction. This class is used to represent a ratio of two integers. Create init, and display methods. Include an additional method, equals, that takes as input another Fraction and returns true if the two fractions are identical and false if they are not. Throw a custom exception if someone tries to create a fraction where the denominator is zero.	Unit -7,8
32	Write a program by creating an 'Employee' class having the following methods and print the final salary. 1 - getInfo() which takes the salary, number of hours of work per day of employee as parameter 2 - AddSal() which adds $10 to salary of the employee if it is less than $500. 3 - AddWork() which adds $5 to salary of employee if the number of hours of work perday is more than 6 hours.	Unit -8
33	Write a program to compare two text files. If they are different, positions in the files where the first difference occurs. Throw custom exceptions if either of the files does not exist. Also throw a custom exception if both files do not have the same content.	Unit -6
34	Create a class called 'Matrix' containing constructor that initializes the number of rows and number of columns of a new Matrix object. The Matrix class has methods for each of the following: 1 - get the number of rows 2 - get the number of columns 3 - set the elements of the matrix at given position (i,j) 4 - adding two matrices. If the matrices are not addable, "Matrices cannot be added" will be displayed. (Overload the addition operator to perform this) 5 - multiplying the two matrices. If the matrices cannot be multiplied, "Matrices cannot be multiplied" will be displayed. (Overload the multiplication operator to perform this) 1 mark for creating appropriate objects of this class and demonstrating all the methods with correct output.	Unit -9
35	Write python code to replicate the following pie chart: For labels: A, B, C, D, E, F and their values: 5,8,9,10,4,7 respectively and their colors blue, green, red, cyan, magenta and yellow respectively. 	Unit -10
36	Create a Point class that consists of an ordered pair (x, y) representing a point's location on the X and Y axes. X and Y coordinates are passed to the constructor on instantiation and default to the origin (0,0) for any missing coordinate. • Overload the + operator to return the sum of two points as another point by adding X and Y coordinates of both points respectively. • Create a method to calculate the slope of line between two points. Formula: $m = \frac{\text{rise}}{\text{run}} = \frac{y_2 - y_1}{x_2 - x_1}$	Unit -8
37	You need to create the foundations of an e-commerce engine for a B2C (business-to-consumer) retailer. You need to have a class for a customer called User, a class for items in inventory called Item, and a shopping cart class called Cart. Items go in Carts, and Users can have multiple Carts. Also, multiple items can go into Carts, including more than one of any single item. Write a program for the above scenario and demonstrate how it works by creating various objects of all classes created.	Unit -8

Major Components/ Equipment	
Sr. No.	**Component/Equipment**
1	Computer
2	Jupyter Notebook

Proposed Theory + Practical Evaluation Scheme by Academicians (% Weightage Category Wise and it's Marks Distribution)					
L :	**2**	**T:**	**0**	**P:**	**6**
Note : In Theory Group, Total 4 Test (T1+T2+T3+T4) will be conducted for each subject. Each Test will be of 25 Marks. Each Test Syllabus Weightage: Range should be 20% - 30%					
Group (Theory or Practical)	**Group (Theory or Practical) Credit**	**Total Subject Credit**	**Category**	**% Weightage**	**Marks Weightage**
Theory	**2**	**5**	MCQ	18%	45
Theory			Theory Descriptive (Mainly Programming)	22%	55
Theory			Formulas and Derivation	0%	0
Theory			Numerical	0%	0
Expected Theory %	**40%**		**Calculated Theory %**	**40%**	**100**
Practical	**3**		Individual Project	13%	25
Practical			Group Project	13%	25
Practical			Internal Practical Evaluation (IPE)	35%	50
Practical			Viva	0%	0
Practical			Seminar	0%	0
Expected Practical %	**60%**		**Calculated Practical %**	**60%**	**100**
Overall %	**100%**			**100%**	**200**

Course Outcome	
1	Students will be able to proficiently write, execute, and debug Python programs using Jupyter Notebooks, utilizing conditional statements, loops, functions, and proper scoping to solve complex problems efficiently.
2	Students will be able to effectively utilize and differentiate between immutable and mutable data structures in Python to manage and manipulate data efficiently.
3	Students will be able to manage files, utilize modules and directories, and apply object-oriented programming principles and exception handling to develop robust Python applications.
4	Students will be able to implement advanced object-oriented programming techniques, perform numerical analysis with NumPy, and generate data visualizations using Matplotlib in Python.
Suggested Reference Books	
1	John V Guttag; Introduction to Computation and Programming Using Python; Prentice Hall
2	Wesley Chun; Core Python Programming; Prentice Hall
3	Kenneth A. Lambert; Fundamentals of Python – First Programs; CENGAGE Publication
4	Robert Johansson; Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib; Apress
5	R. Nageswara Rao; Core Python Programming; dreamtech press

List of Open Source Software/Learning website	
1	https://docs.python.org/3/tutorial/
2	https://numpy.org/doc/stable/
3	https://matplotlib.org/stable/index.html
4	https://www.w3schools.com/python/
5	https://www.coursera.org/specializations/python

Practical Project/Hands on Project		
Sr. No.	**Project List**	**Linked with Unit**
1	Hotel management system : Listed below are some of the important functions **dashboard()** – This function displays the menu or welcome screen to perform different hotel booking activities mentioned below. **new_acc()** – This function creates a new customer account. It asks for some personal and banking details of the customer such as name, date of birth, citizenship number, address and phone number. **room_type()** – This function allows the user to select the categories of the room ie normal or executive with the option of Ac room or non ac room. **check_availability()** – This functionality allows the user to check the number of room vacant prior booking. **book_room()** – This function allows the user to book the selected room. **search_facilities()** – With this function, if the user selects the executive room than user can search for the extra facilities provided like games, swimming, food service in rooms while booking. **payment() –** This function allows making payment of booked room based on number of days the room is occupied via online method option or at the checkout time.	All Units

2	It is required to maintain and process the status of total 9 resources. The status value is to be stored in an integer array of dimension 3x3. The valid status of a resource can be one of the 3 followings: **free**: indicated by integer value 0 **occupied**: indicated by integer value 1 **inaccessible**: indicated by integer value 2 Declare a class called ResourcesStatus, having data member called statusRef, referring to a two dimensional array (3x3) of integers to be used to refer to the above mentioned status values. Define a member method called processStausCount that counts and displays total number of free resources, total number of occupied resources and total number of inaccessible resources. The exception to be raised and handled if total number of occupied resources exceeds total number of free resources. The handler marks status of all inaccessible resources as free. Accept initial status values from user and initialize the array. Raise and handle user defined exception if invalid status value given.	All Units
3	Retail Management System is a simple console-based application. In this project, you can manage a typical 'grocery' department store. You can add goods, edit goods, search, delete and display the goods. Record the information (rate, quantity, name and code) of the added goods. You can search the goods by rate, code or quantity. And, similar goes for display; you can display the items by quantity, rate or code.	All Units
4	Contact Book System, you can manage all contacts. Here, you can add records with name, phone number and the email address. You can view, modify, search and delete existing records. All data added or modified are recorded using and functions and stored I the database Add new contact: add new data with name, phone number and email address View list of contacts – lists all contacts available in the database. Modify contact – edit the added contact; name, phone number and/or email address can be edited Search specific contact – search for a particular contact/updated contact Delete contact – removes contact permanently	All Units
5	Create a two player Rock, Paper, Scissors game using Python. Rules: Rock wins against scissors; paper wins against rock; and scissors wins against paper.	All Units