Faculty of Engineering & Technology
Diploma Engineering (DE)
(W. E. F.: 2023-24)

Document ID: SUTEFETD-01

Name of Faculty	:	Faculty of Engineering & Technology
Name of Program	:	Diploma of Technology (D. Tech)
Course Code	:	2DIT01
Course Title	:	Basics of Python Programming
Type of Course	:	Basic Engineering
Year of Introduction	:	2023-24

Prerequisite	:	High level language (C/C++/Java), Web Programming
Course Objective	:	Develop a strong foundation in Python programming language, including its syntax, data types, and control structures
Course Outcomes	:	At the end of this course, students will be able to:
	CO1	Interpret the fundamental python syntax, semantics and fluent in the use of python control flow statements. Express proficiency in the handling of strings and functions.
	CO2	Apply control structures of python for developing programs
	CO3	Develop a program in Python using built-in functions, modules, and library functions.
	CO4	List and handle exceptions, raise exceptions and create user defined exceptions
	CO5	Determine the methods to create and manipulate python programs by utilizing the data structures like lists, dictionaries, tuples and sets.
	CO6	Develop python programs to solve real world problems

Teaching and Examination Scheme

Teaching Scheme (Contact Hours)			Credits	Examination Marks				
				Theory Marks		Practical Marks		Total Marks
L	**T**	**P**	**C**	**SEE**	**CIA**	**SEE**	**CIA**	
3	0	4	5	70	30	30	20	150

*Legends: **L**-Lecture; **T**–Tutorial/Teacher Guided Theory Practice; **P** – Practical, **C** – Credit, **SEE** – Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.))*

Course Content

Unit No.	Topics	Teaching Hours	Weightage	Mapping with CO
1	**Basics of Python:** Using the Python Interpreter, Variables, Identifiers and Keywords, Numbers and Expressions	02	15%	CO1
2	**Control structures and Function:** Conditional Branching: if Statements, break and continue Statements, and else Clauses on Loops, pass Statements Loops: while Loops, for Loops, Defining Functions, More on Defining Functions: Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations	08	25%	CO2 CO6
3	**Modules and Scoping Rules:** Executing modules as scripts, The Module Search Path, "Compiled" Python files, Packages: Importing * From a Package, Intra-package References, Packages in Multiple Directories	05	20%	CO2 CO6
4	**Exceptions Handling::** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions	07	15%	CO3 CO6
5	Data Structures: Lists, Tuples, Dictionaries and Strings: Common Sequence Operations: Indexing, Slicing, Adding Sequences, Multiplication, Membership, Length, Minimum, and Maximum, Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, the del statement, Tuples and Sequences, Sets, Dictionaries, Comparing Sequences and Other Types, Basic String Operations	08	25%	CO4 CO6

Suggested Distribution of Theory Marks Using Bloom's Taxonomy						
Level	Remembrance	Understanding	Application	Analyse	Evaluate	Create
Weightage	**20**	**35**	**35**	**0**	**0**	**0**

NOTE: This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Suggested List of Experiments/Tutorials

Sr. No.	Name of Experiment/Tutorial	Teaching Hours
1	Create a program that asks the user to enter their name and their age. Printout a message addressed to them that tells them the year that they will turn 100 years old.	02
2	Ask the user for a number. Depending on whether the number is	02
3	even or 2 odd, print out an appropriate message to the user. Hint: how does an even / odd number react differently when divided by 2?	02
4	Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)	02
5	Take two lists, say for example these two: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.	02
6	Ask the user for a string and print out whether this string is a palindrome or not. (A palindrome is a string that reads the same forwards and backwards.)	02
7	Let's say I give you a list saved in a variable: a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list and makes a new list that has only the even elements of this list in it.	02
8	Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game) Remember the rules: Rock beats scissors, Scissors beats paper, Paper beats rock	02
9	Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right. (Hint: remember to use the user input lessons from the very first practical)	02
10	This week's exercise is going to be revisiting an old exercise (see Practical 3), except require the solution in a different way. Take two lists, say for example these two: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes. Write this in one line of Python using at least one list comprehension	02
11	Ask the user for a number and determine whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.). You can (and should!) use your answer to Practical 2 to help you. Take this opportunity to practice using functions, described below.	02

12	Write a program that takes a list of numbers (for example, a = [5, 10, 15, 20, 25]) and makes a new list of only the first and last elements of the given list. For practice, write this code inside a function.	02
13	Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate. (Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, …)	02
14	Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.	02
15	Write a program (using functions!) that asks the user for a long string function. Containing multiple words. Print back to the user the same string, except with the words in backwards order. For example, say I type the string: My name is Michele Then I would see the string: Michele is name My shown back to me.	02
16	Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password. Include your run-time code in a main method.	02
17	Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.	02
18	Python supports classes inheriting from other classes. The class being inherited is called the Parent or Superclass, while the class that inherits is called the Child or Subclass. How can we define the order in which the base classes are searched when executing a method?	02
19	Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate boolean.	02
20	Given a .txt file that has a list of a bunch of names, count how many of each name there are in the file, and print out the results to the screen.	02
21	Write a Program to remove all whitespaces using regular expressions	02
22	Write a program to accept 10 numbers from the user and sort the 10 numbers using bubble sort and insertion sort.	02
23	Write a program to catch on Divide by zero Exception with finally clause.	02
24	Write a user-defined exception that could be raised when the text entered by a user consists of less than 10 characters.	02
25	Write a python program to demonstrate exception handling.	02

Major Equipment/ Instruments and Software Required

Sr. No.	Name of Major Equipment/ Instruments and Software
1	Python IDLE
2	Anaconda Python

3	PyCharm

Suggested Learning Websites

Sr. No.	Name of Website
1	https://www.python.org/
2	http://www.diveintopython3.net/
3	https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django
4	https://www.fullstackpython.com/django.html

Reference Books

Sr. No.	Name of Reference Books
1	David Beazley, Brian K. Jones, "Python Cookbook", 3rd edition, OREILLY,2016
2	Brett Slatkin, "Effective Python: 59 Specific Ways to Write Better Python", Novatec, 2016
3	Allen Downey, "Think Python: How to Think Like a Computer Scientist", Green Tea Press,2015
4	Mark Lutz "Learning Python", 4th Edition, O'REILLY, 2016
5	Arun Ravindran, Aidas Bendoraitis, Samuel Dauzon, "Django: Web Development with Python",Packt Publishing, 2016