



Charotar University of Science and Technology
Devang Patel Institute of Advance Technology and Research
(DEPSTAR)
Department of Computer Engineering
Department of Computer Science & Engineering



Subject: Programming in Python

Semester: 4

Subject Code: CE259

Academic Year: 2021-22

Course Outcome (COs):

At the end of the course, the students will be able to

1. 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.
2. Determine the methods to create and manipulate python programs by utilizing the data structures like lists, dictionaries, tuples and sets.
3. Identify the commonly used operations involving file systems and regular expressions.
4. Articulate the Object-Oriented Programming concepts such as encapsulation, inheritance and polymorphism as used in Python along with magic methods.

Practical List

Sr. No.	AIM	Hrs	COs
Pre Req1	Introduction to Python Programming. Installation & Configuration of Python. Along with its all major editors, IDLE, PyCharm, Anaconda, Jupyter, Interpreter etc.	1	1,2
1	1.1 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.	2	1,2
	1.2 Write a program to ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user. Hint: how does an even / odd number react differently when divided by 2?		1,2
2	2.1 Write a program as mentioned below: Take a list, a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89], and write a program that print out all the elements of the list that are less than 5.	2	2,3
	2.2 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.)		2,3
3	3.1 Take two lists, 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.	2	2,3
	3.2 Write a program by asking 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.)		1,2
4	4.1 Write one line of Python that takes list a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100] and makes a new list that has only the even elements of this list in it.	2	2,3
	4.2 Write a program to 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) Rules: Rock beats scissors, Scissors beats paper, Paper beats rock		1,2
5	5.1 Write a program to 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)	2	1,2
	5.2 Take two lists, 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		1,2
6	6.1 Write a program using a function to check whether the number is prime or not. (A prime number is a number that has no divisors.)	2	1,2
	6.2 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.		2,3



Charotar University of Science and Technology
Devang Patel Institute of Advance Technology and Research
(DEPSTAR)
Department of Computer Engineering
Department of Computer Science & Engineering



7	7.1 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, ...)	2	1,2
	7.2 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.		1,2
8	8.1 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.	4	2,3
	8.2 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.		1,2
9	9.1 Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.	4	2,4
	9.2 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.		2,4
10	10.1 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.	4	3,4
	10.2 Write a program to implement different Data structures using Python. <ul style="list-style-type: none"> • Linked List • Stack • Queue • Binary Tree 		1,4
11	Develop programs to understand the working of exception handling with the user guessing a number until he/she gets it right.	2	1,3
12	12.1 Create two 2D Numpy arrays with random numbers and concatenate them using the Numpy library. After Concatenation, reshape the resulting Numpy array such that the number of rows and columns is reversed.	2	2,4
	12.2 Create a Pandas series from a Python List. Find out the mean, median, mode, range and standard deviation of the series.		2,4