CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

FACULTY OF TECHNOLOGY & ENGINEERING

SMT. KUNDANBEN DINSHA PATEL DEPARTMENT OF INFORMATION TECHNOLOGY

Subject Name: Python Programming **Semester: B.Tech V**
Subject Code: IT374 **Academic year: June -Dec 2019**

Note: The laboratory will emphasize the Python Programming (basic to advance), use of Python Packages and libraries.
Instructions:

1. **All Practical must be performed individually and all experimental results must be uploaded on your respective EDMODO account**.
2. All Practical must be evaluated regularly in the laboratory by concern Lab Teacher.
3. Each practical answer would be evaluated as learning outcome.

Practical List

Sr. No.	Aim of the Practical	Hrs
Pre Req1	Introduction to Python Programming. Installation & Configuration of Python. Along with its all major editors, IDLE, Pycharm, Anaconda, Jupyter, Interpreter etc.	-
Pre Req2	Every student has to create **CodeZinger account**, join the course named "Introduction to Python" and solve the given exercise as per the instructions. Which you will get from your lab faculty member.	-
1.	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.	2
2.1	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? **Self:** If the number is a multiple of 4, print out a different message.	2
2.2	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. **Self:** Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.	2
3.1	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
3.2	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 using at least one list comprehension. **Self:** Randomly generate two lists to test this	2

3.3	Ask the user for a string and print out whether this string is a palindrome or not. (Using string reversal and for loops).	2
4.1	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. **Self:** Create list contains square of all odd numbers from range 1 to 10 Create list contains power of 2 from 1 to 8 Create list contains prime and non-prime in range 1 to 50	4
4.2	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	2
4.3	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. **Self:** Keep the game going until the user types "exit"	2
5.1	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.	2
5.2	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
5.3	Write a Python program to print all unique values in a dictionary. Sample Data : [{"V":"S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"}, {"VII":"S005"}, {"V":"S009"},{"VIII":"S007"}] Expected Output : Unique Values: {'S005', 'S002', 'S007', 'S001', 'S009'} **Self:** Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary. Go to the editor Sample data : {'1':['a','b'], '2':['c','d']} Expected Output: ac, ad, bc, bd	4
6.1	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
6.2	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. **Self:** Write two different functions to do this - one using a loop and constructing a list, and another using sets.	4

6.3	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.	2
7.1	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. **Self:** Ask the user how strong they want their password to be. For weak passwords, pick a word or two from a list.	4
7.2	Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.	2
8.1	Implement Multiple Inheritance. 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? (Hint: MRO: Method Resolution Order)	2
8.2	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
8.3	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
9.1	Develop programs to learn regular expressions using python.	2
9.2	Develop programs for data structure algorithms using python – sorting (Bubble sort and Insertion sort)	2
9.3	Develop programs to understand working of exception handling and assertions.	2
10	Introduction to Django- Python based free and open-source web framework and Flask-Python based micro web framework.	4