

# **IT-Workshop using Python (PCC-CS 393)**

## **Laboratory Instructor's Manual**



**Last Revised**

August, 2022

Dept. of CSE  
Techno Main, Salt Lake



### **GENERAL INSTRUCTIONS FOR STUDENTS**

1. Do not enter the Laboratory without prior permission.
2. Switch off your mobile phones during Lab class and maintain silence.
3. Save your files only on the specific destination folders as instructed.
4. Do not play games, watch movies, chat or listen to music during the class.
5. Do not change desktop setting, screen saver or any other system settings.
6. Do not use any external storage device without prior permission.
7. Do not install any software without prior permission.
8. Do not browse any restricted, illegal or spam sites.

### **GENERAL ADDRESS FOR LABORATORY TEACHERS**

1. Submission of documented lab reports related to completed lab assignments should be done during the following lab session.
2. The promptness of submission should be encouraged by way of marking and evaluation patterns as reflected in the lab rubric which eventually will benefit the students.

### **Program Outcomes (PO)**

**PO1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and engineering specialization to the solution of complex engineering problems.

**PO2. Problem analysis:** Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using first principles of mathematics, natural and engineering sciences.

**PO3. Design/Development of solutions:** Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety and the cultural societal and environmental considerations.

**PO4. Conduct investigations of complex problems:** Use research based knowledge including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

**PO5. Modern tool usage:** Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6. The engineer and society:** Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.

**PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9. Individual and team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

**PO10. Communications:** Communicate effectively with the engineering community and with the society at large. Be able to comprehend and write effective reports documentation. Make effective presentations and give and receive clear instructions.

**PO11. Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

**PO12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.



### **Program Specific Outcomes (PSOs)**

- PSO1:*** Ability to develop the solutions for scientific, analytical and research-oriented problems in the area of Computer Science and Engineering.
- PSO2:*** Ability to apply suitable programming skills integrated with professional competence to develop applications catering to the industrial and societal needs in the field of Computer Science and Engineering and its allied areas.



NAME OF THE PROGRAM: <b>CSE</b>	DEGREE: <b>B.Tech</b>
COURSE NAME: <b>IT-Workshop (using Python)</b>	SEMESTER: <b>3rd</b>
COURSE CODE: <b>PCC-CS393</b>	COURSE CREDIT: <b>3</b>
COURSE TYPE: <b>PRACTICAL</b>	CONTACT HOURS: <b>4P</b>

## **SYLLABUS**

### **a. Introduction**

History, Features, Setting up path, Working with Python, Basic Syntax, Variable and Data Types, Operator

### **b. Conditional Statements**

If, If- else, Nested if-else, Looping, For, While, Nested loops

### **c. Control Statements**

Break, Continue, Pass

### **d. String Manipulation**

Accessing Strings, Basic Operations, String slices, Function and Methods

### **e. Lists**

Introduction, Accessing list, Operations, Working with lists, Function and Methods

### **f. Tuple**

Introduction, Accessing tuples, Operations, Working, Functions and Methods

### **g. Dictionaries**

Introduction, Accessing values in dictionaries, Working with dictionaries, Properties

### **h. Functions**

Defining a function, Calling a function, Types of functions, Function Arguments, Anonymous functions, Global and local variables

### **i. Modules**

Importing module, Math module, Random module, Packages, Composition, Input-Output  
Printing on screen, Reading data from keyboard, Opening and closing file, Reading and writing files, Functions

### **j. Exception Handling**

Exception, Exception Handling, Except clause, Try? finally clause, User Defined Exceptions.



NAME OF THE PROGRAM: <b>CSE</b>	DEGREE: <b>B.Tech</b>
COURSE NAME: <b>IT-Workshop (using Python)</b>	SEMESTER: <b>3rd</b>
COURSE CODE: <b>PCC-CS393</b>	COURSE CREDIT: <b>3</b>
COURSE TYPE: <b>PRACTICAL</b>	CONTACT HOURS: <b>4P</b>

### **Course Outcome (CO)**

*After this course students will be able to:*

- CO-1:** Make use of basic Python language constructs and functions.
- CO-2:** Build simple applications using various Python modules and debug Python programs.
- CO-3:** Apply basic concepts of Image Processing, Machine Learning, etc. with the knowledge of Python to write useful applications.
- CO-4:** Develop Python programs as an individual or in a group and prepare appropriate documentation.

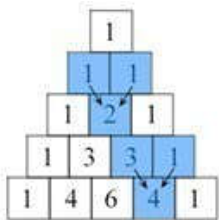
NAME OF THE PROGRAM: <b>CSE</b>	DEGREE: <b>B.Tech</b>
COURSE NAME: <b>IT-Workshop (using Python)</b>	SEMESTER: <b>3rd</b>
COURSE CODE: <b>PCC-CS393</b>	COURSE CREDIT: <b>3</b>
COURSE TYPE: <b>PRACTICAL</b>	CONTACT HOURS: <b>4P</b>

Exp. No.	Experiments	CO Mapping	Week
1.	<p>a. Write a Python program to add two numbers.</p> <p>b. Write a Python program to find remainder when a number is divided by z.</p> <p>c. Check the type of the variable assigned using input () function.</p> <p>d. Use comparison operator to find out whether a given variable 'a' is greater than 'b' or not. For example, take a=34,b=80.</p> <p>e. Write a Python program to find the average of two numbers entered by the users.</p> <p>f. Write a Python program to calculate square of a number entered by the user.</p> <p><b>NOTE:</b> For all the programs try to take user input for at least 3 cases.</p>	CO1 CO2 CO4	Week 1
2.	<p>a. Write a program to print the first 10 even numbers in reverse order.</p> <p>b. Write a program to print a table of a number accepted from the user.</p> <p>c. Write a program to find the factorial of a number.</p> <p>d. Write a program to print all prime numbers that fall between two numbers including both (accept two numbers from the user)</p> <p>e. Write a program that keeps on accepting numbers from the user until the user enters Zero. Display the sum and average of all the numbers.</p> <p>f. Write a program to print the following pattern</p> <pre> 5 5 5 5 4 4 4 4 3 3 3 2 2 1 </pre> <p><b>NOTE:</b> For all the programs try to take user input for at least 3 cases.</p>	CO1 CO2 CO4	Week 2

Exp. No.	Experiments	CO Mapping	Week
3.	a. Write a Python Program to insert a string in the middle of a string b. Write a Python program to get a string made of 4 copies of the last two characters of a specified string (length must be at least 2). c. Write a python program to check the string is anagram or not. d. Write a Python program to sort a string lexicographically. e. Write a Python program to add a prefix text to all of the words in a string.	CO1 CO2 CO4	Week 3
4.	a. Write a Python program to interchange the first and last elements in a list. b. Write a Python program to Reverse a List. c. Write a Python program to count Even and Odd numbers in a List. d. Write a Python program to find second largest number in a list. e. Write a Python program to count positive and negative numbers in a list.	CO1 CO2 CO4	Week 4
5.	a. Write a Python program to Swap two tuples in Python. b. Write a Python program to Copy specific elements from one tuple to a new tuple c. Write a python program to Add Tuple to List and vice – versa. d. Input two tuples and write a python program to find Modulo of tuple elements and store in a third tuple. e. Write a Python program to create a list of tuples from given list having number and its cube in each tuple. (e.g. (2,8),(3,27),....).	CO1 CO2 CO4	Week 5
6.	a) Write a Python program to create a dictionary and do the following: <ul style="list-style-type: none"> <li>i) Sort Dictionary by key.</li> <li>ii) Sort Dictionary by value.</li> </ul> b) Write a Python Program to create grade calculator, creating a dictionary which consists of the student name, assignment result test results and their respective lab results.  Given different scored marks of students. We need to find grades. The test score is an average of the respective marks scored in assignments, tests and lab-works. The final test score is assigned using below formula.	CO1 CO2 CO4	Week 6



Exp. No.	Experiments	CO Mapping	Week
	<p>10 % of marks scored from submission of Assignments</p> <p>70 % of marks scored from Test</p> <p>20 % of marks scored in Lab-Works</p> <p>Grade will be calculated according to :</p> <p>score <math>\geq</math> 90 : "A"</p> <p>score <math>\geq</math> 80 : "B"</p> <p>score <math>\geq</math> 70 : "C"</p> <p>score <math>\geq</math> 60 : "D"</p> <p>Example input:</p> <pre>james = { "name": "James Potter",           "assignment" : [82, 56, 44, 30],           "test" : [80, 80],           "lab" : [67.90, 78.72]         }</pre> <p>Example output:</p> <p>Average marks of James Potter is : 75.962</p> <p>Letter Grade of James Potter is : C</p> <p>c) Write a program to create two Dictionaries. Take a key from the 1st dictionary and find it in the 2nd dictionary. Display the corresponding value if found otherwise print appropriate message.</p> <p>Also merge the two dictionaries and display the output.</p> <p>d) Write a python program to Replace words from Dictionary.</p> <p>e.g. Input : test_str = 'TechnoIndia is one of the best colleges in India', repl_dict = {"India": "West Bengal"}</p> <p>Output : TechnoIndia is one of the best colleges in West Bengal</p> <p>Explanation : "India" word is replaced by lookup value.</p>		Week 6

Exp. No.	Experiments	CO Mapping	Week
7.	<p>a) Write a Python function to find the Max of threenumbers.</p> <p>b) Write a Python function to multiply all the numbers in alist.</p> <p style="padding-left: 40px;">Sample List: [8, 2, 3, -1, 7]</p> <p style="padding-left: 40px;">Expected Output: -336</p> <p>c) Write a Python function that accepts a string and calculatethe number of upper case letters and lower caseletters.</p> <p style="padding-left: 40px;">Sample Input: 'The quick Brown Fox'</p> <p style="padding-left: 40px;">Expected Output:</p> <p style="padding-left: 40px;">No. of Upper case characters: 3</p> <p style="padding-left: 40px;">No. of Lower case Characters: 12</p> <p>d) Write a Python function that prints out the first n rows ofPascal's triangle.</p> <p style="padding-left: 40px;">Note: Pascal's triangle is an arithmetic and geometric figure first imagined by Blaise Pascal.</p> <p style="padding-left: 40px;">Sample Pascal's triangle:</p> <div style="text-align: center;">  </div> <p style="padding-left: 40px;">Each number is the two numbers above it added together.</p> <p>e) Write a Python program, where N number of integer arguments are passed to a function make_sum(), which will print the sum of all the passed integers.</p>	CO1 CO2 CO4	Week 7

Exp. No.	Experiments	CO Mapping	Week
8.	<p>a) Write a Python program to read first n lines of a file</p> <p>b) Write a Python program to count the number of lines in a text file.</p> <p>c) Write a Python program to write a list to a file.</p> <p>d) Write a Python program to copy the contents of a file to another file.</p> <p>e) Write a Python program to take a file name and a string pattern from the user. Now print the number of occurrence of the pattern string found in the file.</p>	CO1 CO2 CO4	Week 8
9.	<p>a) Write a Python program to generate a random alphabetical string, random value between two integers (inclusive) and a random multiple of 7 between 0 and 70.</p> <p>Generate a random alphabetical string:</p> <p>lGhPpBDqfCgXKzSbGcnmcDWBEZeiqcUqztgvwcXfVyPslOggKdbI xOejJfFMgspqrskanNYpscJEOVIpYkGGNxQlaqeubGDbQSBhBedr dOyqOmKPTZvzKmKVOIDs</p> <p>Generate a random value between two integers, inclusive:</p> <p>0</p> <p>4</p> <p>1</p> <p>Generate a random multiple of 7 between 0 and 70:</p> <p>70</p> <p>b) Write a Python program to generate a random integer between 0 and 6 - excluding 6, random integer between 5 and 10 - excluding 10, random integer between 0 and 10, with a step of 3 and random date between two dates.</p> <p>Generate a random integer between 0 and 6:</p> <p>0</p> <p>Generate random integer between 5 and 10, excluding 10:</p> <p>5</p> <p>Generate random integer between 0 and 10, with a step of 3:</p> <p>3</p>	CO1 CO2 CO4	Week 9

Exp. No.	Experiments	CO Mapping	Week
	<p>Random date between two dates:</p> <p>2019-02-17</p> <p>c) Write a Python program to convert degree to radian.</p> <p>Note: The radian is the standard unit of angular measure, used in many areas of mathematics. An angle's measurement in radians is numerically equal to the length of a corresponding arc of a unit circle; one radian is just under 57.3 degrees (when the arc length is equal to the radius).</p> <p>Test Data:</p> <p>Degree: 15</p> <p>Expected Result in radians: 0.2619047619047619</p> <p>d) Write a Python program to calculate surface volume and area of a cylinder.</p> <p>Note: A cylinder is one of the most basic curvilinear geometric shapes, the surface formed by the points at a fixed distance from a given straight line, the axis of the cylinder.</p> <p>Test Data:</p> <p>volume: Height (4), Radius (6)</p> <p>Expected Output:</p> <p>Volume is: 452.57142857142856</p> <p>Surface Area is: 377.1428571428571</p>		Week 9
10.	<p>a) Below, we have provided a buggy code. Add a try/except clause so the code runs without errors. If a blog post didn't get any likes, a 'Likes' key should be added to that dictionary with a value of 0.</p> <pre>blog_posts = [{ 'Photos': 3, 'Likes': 21, 'Comments': 2 }, { 'Likes': 13, 'Comments': 2, 'Shares': 1 }, { 'Photos': 5, 'Likes': 33, 'Comments': 8, 'Shares': 3 }, { 'Comments': 4, 'Shares': 2 }, { 'Photos': 8, 'Comments': 1, 'Shares': 1 }, { 'Photos': 3, 'Likes': 19, 'Comments': 3 }]</pre> <p>total_likes = 0</p> <p>for post in blog_posts:</p> <p style="padding-left: 40px;">total_likes = total_likes + post['Likes']</p>	CO1 CO2 CO4	Week 10

Exp. No.	Experiments	CO Mapping	Week
	<p>b) The code below assigns the 5th letter of each word in 'food' to the new list fifth. However, the code currently produces errors. Insert a try/except clause that will allow the code to run and produce a list of the 5th letter in each word. If the word is not long enough, it should not print anything out. Note: The 'pass' statement is a null operation; nothing will happen when it executes.</p> <pre> food = ["chocolate", "chicken", "corn", "sandwich", "soup", "potatoes", "beef", "lox", "lemonade"]  fifth = []  for x in food:      fifth.append(x[4]) </pre> <p>c) In Python, write an interactive calculator. Take input as command line arguments, which is assumed to be a formula that consist of a number, an operator (at least + and -), and another number, separated by white space (e.g. 1 + 1). Check whether the input arguments are valid before computing the result:</p> <p>If the input does not consist of 3 elements, raise a FormulaError, which is a custom Exception.</p> <p>Try to convert the first and third input to a float (like so: float_value = float(str_value)). Catch any ValueError that occurs, and instead raise a FormulaError</p> <p>If the second input is not '+' or '-', again raise a FormulaError</p> <p>If the input is valid, perform the calculation and print the result, else print appropriate error messages and quit.</p> <p>d) Imagine you have a file named data.txt. Open it for reading using Python code, but make sure to use a try block to catch an exception that arises if the file doesn't exist. Once you've verified your solution works with an actual file, delete the file and see if your try block is able to handle it.</p> <p><b>Note</b> that the exception we need to watch out for is FileNotFoundError.</p>		Week 10
11.	a). Write a Python script to read a gray-scale image and display its Histogram.	CO1 CO2 CO3 CO4	Week 11

Exp. No.	Experiments	CO Mapping	Week
	<p>b) Write a Python script to enhance the image used in Question No. 1 using simple Histogram Equalization technique. Also display the equalized histogram along with the enhanced image.</p> <p>Submit the following,</p> <p>i) program code (.py) for Q.1</p> <p>ii) original image file</p> <p>iii) screenshot of histogram of originalimage</p> <p>iv) program code (.py) for Q.2</p> <p>v) enhanced image file</p> <p>vi) screenshot of histogram of the enhanced image</p>		Week 11
12.	Write a script in Python for Normal Data Point clustering.	CO1 CO2 CO3 CO4	Week 12



NAME OF THE PROGRAM: <i>CSE</i>	DEGREE: <i>B.Tech</i>
COURSE NAME: <i>IT-Workshop (using Python)</i>	SEMESTER: <i>3rd</i>
COURSE CODE: <i>PCC-CS393</i>	COURSE CREDIT: <i>3</i>
COURSE TYPE: <i>PRACTICAL</i>	CONTACT HOURS: <i>4P</i>

### Rubrics for Lab

Criteria \ Score	Excellent (10-8)	Good (7-6)	Average (5-4)	Poor (3-1)	CO Mapping	PO/PSO Mapping
<b>Lab Participation (Following Procedure +Lab Techniques+ Subject Knowledge + Contribution)</b>	Student demonstrates an accurate understanding of the lab assignments. The student can correctly answer questions and if appropriate, can explain concepts to fellow classmates. Student is eager to develop new ideas and assists when needed.	Student arrives on time to lab, but may be underprepared. Answers to questions are basic and superficial suggesting that concepts are not fully grasped. Able to follow the instruction and somehow managed to execute the program.	Student unpreparedness makes it impossible to fully participate. If able to participate, student has difficulty explaining key lab concepts.	There was no attempt to make prior arrangements to make up the lab. Attendance is not regular. Not able to run the program even after getting help from the peers.	<b>CO1, CO2, CO3, CO4</b>	<b>PO1 / PSO2</b>
<b>Interaction with Group (Team work)</b>	Very good participation with a good leadership quality; is respectful of others and their point of view; makes sure that everyone gets a turn; conscious of time	Good participation; appears interested; enthusiastic but talks over teammates; try to help group complete tasks; somewhat conscious of time	Minimal participation; shows little interest; doesn't pay attention to other group members; may argue to get point across; helps group only when asked; little emphasis on time	No participation; sits on the sidelines with no interaction; disinterested; no stake in time management	<b>CO2, CO3, CO4</b>	<b>PO1, PO2, PO3 / PSO2</b>



Criteria \ Score	Excellent (10-8)	Good (7-6)	Average (5-4)	Poor (3-1)	CO Mapping	PO/PSO Mapping
<b>Execution and Debugging (Modern tool usage)</b>	Follow the logical ideas; can develop suitable program from specific algorithm; debug the program with proficiency; Able to check the reliability	Can develop suitable program from specific algorithm with the help of the instructor; debug the program with proficiency; Able to check the reliability	Can develop suitable program from specific algorithm with the help of the instructor; debug the program with the help of technical assistant; Not able to check the reliability.	Not be able to develop suitable program from specific algorithm; need assistance to debug the program. Not able to check the reliability	<b>CO1, CO2, CO3</b>	<b>PO1, PO5 / PSO2</b>
<b>Lab Report</b>	Student demonstrates an accurate understanding of the lab concepts. Questions are answered completely and correctly. Output of each program is neat, creative and includes complete titles. Errors, if any are minimal	Student has a basic knowledge of content, but may lack some understanding of some concepts. Questions are answered fairly well and/or output could have been done more neatly, accurately or with more complete information	Student has problems with both the output and the answers. Student appears to have not fully grasped the lab content and the code possess multiple errors	Student turns in lab report late or the report is so incomplete and/or so inaccurate that it is unacceptable.	<b>CO4</b>	<b>PO1, PO9, PO10 / PSO2</b>