



Faculty of Computer Applications & Information Technology
Integrated Master of Computer Applications [iMCA]
Semester V
PRACTICALS ON PYTHON
(Core Course)

Course Code: 222301506

Course Credit: 3

• **Introduction:**

Python Practical involves hands-on experience and application of Python programming in real-world scenarios. Students or learners work on projects that allow them to build functional software or solve practical problems using Python. Python is known for its versatility, making Python Practical a valuable skill for various professions and applications.

• **Course Objectives:**

1. Understand the fundamental programming concepts including types, variables, branching and loops.
2. Students will be able to work with module, objects and classes.
3. Students will learn how to use and manipulate several core data structures: Lists, Dictionaries, Tuples, and Strings.
4. Students will be able to work with GUI in python.

• **Course Duration:**

The course will have sessions which are divided into five modules. Each module consists of nine sessions of 60 minutes each and carries a weightage of 20%.

• **Course Contents:**

Module No.	Modules/Sub-Modules	No. of Sessions	Marks Weightage
I	Practicals related to Python Basic Concepts <ul style="list-style-type: none">• Write a program to swap of two numbers.• Write a program to convert the temperature from degree centigrade to Fahrenheit. $[C = ((F-32) * 5)/9]$• Write a program to calculate the sum of 5 subject and find the percentage.• Write a program to check whether the variable is of type integer or float. (Note: Use Isinstance method).• Write a program to get the Python version.• Write a program to display the current date and time.• Write a program to demonstrate the mathematical constant pi and e.• Write a program to demonstrate the use of random function.• Write a program to convert numbers into string.• Write a program to demonstrate use of subscript operator for accessing range wise substring.<ul style="list-style-type: none">• Example: print "var2[1:5]: ", var2[1:5], var2 = "Python Programming"• Write a program to demonstrate the use of following methods<ul style="list-style-type: none">• abs, ceil, cmp, exp, round• floor, max, min, power	9	20%

	<ul style="list-style-type: none"> • Write a program to find square root of a number using math.sqrt() function • Write a program to update string. • Write a program to demonstrate use of str.capitalize() method. • Write a program to demonstrate use of str.center() method. • Write a program to find length of string. • Write a program to find length of substring from a string. • Write a program to compare two string variables. • Write a program to concatenate two string variables. • Write a program to demonstrate use of subscript operator for accessing substring. Example: var1='Hello World!', print "var1[0]: ", var1[0] • Write a program to demonstrate the mathematical constant pi and e. • Write a program to demonstrate the use of random function. • Write a program to convert numbers into string. • Write a program to replace a string from a sentence. • Write a program to demonstrate the use of following methods <ul style="list-style-type: none"> • Endswith, index, join, len, lower, startswith • isalnum, isalpha, isdigit, islower, translate • isnumeric, isspace, istitle, isupper, swapcase • max, min, rfind, upper, split, splitlines, isdecimal 		
II	<p>Practicals related to Programming Constructs and Data Structures</p> <ul style="list-style-type: none"> • Try following code in python <pre>max = (a > b) ? a : b; if (a > b) max=a; else max=b; max = a if (a > b) else b;</pre> • Use for loop to display each word in python. words = ['This','is','Python','For','Loop','Example'] • Write a program to demonstrate the use of else in while loop • Write a program to demonstrate the use of range in for loop. • Write a program to find factorial of a number. • Calculate the Basic Salary of an employee where: HRA is 10% of Basic Salary, DA is 20% of Basic Salary TA is 10% of Basic Salary PF is 20 % of Basic Salary. Also calculate the net salary of the employee. • Write a program to check whether the blood donor is eligible or not for donating blood. The conditions laid down are as under. Use if statement. <ul style="list-style-type: none"> • Age should be above 18 yrs. but not more than 55 yrs. • Weight should be more than 45kgs. • Write a program to print following character triangle: <pre> A A B A A B C B A A B C D C B A A B C D E D C B A</pre> • Write a program to Check Armstrong Number • Write a Program to Find the Sum of first 50 Natural Numbers using For Loop. • Write a program to Display Factors of a Number • Write a Program to Check Whether a Number is Palindrome or 	9	20%

	<p>Not</p> <ul style="list-style-type: none"> • Write a Program to Check Whether a Number is Prime or Not using function • Write a Program to Reverse a Number • Write a program to Calculate the Power of a Number • C program to Generate Multiplication Table. • Write a Python program to multiply all the items in a list. • Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples. Sample List: [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)] Expected Result: [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)] • Write a Python program to create a list by concatenating a given list with a range from 1 to n. Sample list: ['p', 'q'] n =5 Sample Output: ['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5'] • Write a Python program to get the 4th element from the last element of a tuple. • Write a Python program to print a tuple with string formatting. Sample tuple: (100, 200, 300) Output: This is a tuple (100, 200, 300) • Write a Python program to convert a given string list to a tuple. Original string: python 3.0 <class 'str'> Convert the said string to a tuple: ('p', 'y', 't', 'h', 'o', 'n', '3', '.', '0') <class 'tuple'> • Write a Python script to check whether a given key already exists in a dictionary. • Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are the square of the keys. Sample Dictionary {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225} • Write a Python program to combine two dictionaries by adding values for common keys. d1 = {'a': 100, 'b': 200, 'c': 300} d2 = {'a': 300, 'b': 200, 'd': 400} Sample output: Counter ({'a': 400, 'b': 400, 'd': 400, 'c': 300}) • Write a Python program to remove an item from a set if it is present in the set. • Write a Python program to create an intersection, union and difference of sets. • Write a Python program that uses frozensets. Note: Frozensets behave just like sets except they are immutable. 		
III	<p>Practical related to Python Functions, Modules and Packages</p> <ul style="list-style-type: none"> • Write a Program to show Fibonacci Series with function. • Write a Program to Display sum of two numbers using function. • Write a program to find square of a number using functions. • Write a Python program to generate a random color hex, a random 	9	20%

	<p>alphabetical string, random value between two integers (inclusive) and a random multiple of 7 between 0 and 70. Use random.randint()</p> <ul style="list-style-type: none"> • Write a Python program to check if a function is a user-defined function or not. Use types.FunctionType, types.LambdaType() • Create a Python module named basic_math.py that contains functions for basic mathematical operations such as addition, subtraction, multiplication, and division. • Create a Python module named user_auth.py that contains functions for user authentication. • Create a package named geometry that contains modules for basic geometric operations such as calculating the area and perimeter of shapes like squares, rectangles, and circles. • Create a Python package named Financial that contains modules for performing various financial calculations such as simple interest, compound interest, present value, and future value. 		
IV	<p>Practicals related to Classes and Exceptions</p> <ul style="list-style-type: none"> • Create a class named Calculator that performs basic arithmetic operations like addition, subtraction, multiplication, and division. • Create a class named ToDoList that represents a simple todo list and perform operations like add task, remove task. • Create a class named Library to represent a simple book library and implement methods like add book, remove book, display book. • Write a Python program to handle a ZeroDivisionError exception when dividing a number by zero. • Write a Python program that prompts the user to input two numbers and raises a TypeError exception if the inputs are not numerical. • Write a Python program that opens a file and handles a FileNotFoundError exception if the file does not exist. • Write a Python script that prompts the user to enter two numbers and an operator (+, -, *, /). Perform the corresponding arithmetic operation and handle exceptions for invalid inputs such as non-numeric operands or unsupported operators. • Write a Python script that prompts the user to enter a list of numbers separated by spaces. Then prompt the user to enter an index and attempt to access the element at that index in the list. Handle the IndexError exception if the specified index is out of range. 	9	20%
V	<p>Practicals related to Python GUI - Tkinter</p> <ul style="list-style-type: none"> • Create a GUI Application to demonstrate working of calculator using Tkinter. • Create a Student Registration form for library. • Create a GUI Application to set title of a window. • Create a GUI Application to set the size of a window. • Create a simple GUI Application to demonstrate your College Name in label. • Create a GUI Application to count Button Click. • Create a GUI Application to find odd/even on Button Click. • Create a GUI Application to display Movie Chooser using check box control. • Create a GUI Application to display Hobby Chooser using check box control. • Create a Student Registration form for college. 	9	20%

<ul style="list-style-type: none"> • Create a GUI Application to find odd/even on radio Button Click. • Create a GUI Application to find positive/negative on radio Button Click. <p>Practical's related to Python Mysql</p> <ul style="list-style-type: none"> • Write a Python script to insert a new employee record into a MySQL table named employees. The table has columns: id , name, position, and salary. • Insert multiple records into the employees table in a single query. • Create a Python function that updates the salary of an employee with a specific id in the employees. • Write a Python script to delete an employee record from the employees table based on the employee's id. • Write a Python script that inserts a new product into a MySQL table named products. The table has columns: product_id (auto-incremented), product_name, price, and stock_quantity. • Write a Python script to demonstrate use of python libraries (Numpy, Pandas, Scipy, Matplotlib) 	
--	--

• **Course Outcomes (CO):**

- CO1: Understand the fundamental concepts of Python programming, including variables, data types, and control structures.
- CO2: Analyzing and solving exceptions and errors
- CO3: Understanding the different data structure in python
- CO4: Using the basics of classes and objects in Python
- CO5: Explore python library and built-in modules
- CO6: Designing and implementing GUI applications with Tkinter

• **Teaching Scheme:**

Teaching Scheme (Hours): 45		
Course Credit: 3		
Theory	Tutorial	Practical
-	-	45

• **Examination Scheme:**

Examination Scheme				
External	Internal	CEC		Total Marks
		Assignments	(Presentations, Group Discussion, Projects, Quiz, Viva, Education Tours)	
50%	20%	10%	20%	100%

• **Teaching Methods:**

The following pedagogical tools will be used to teach this course:

1. Lectures and Discussions
2. Assignments and E-Learning
3. Practical Demos

• **Supplementary Readings:**

1. Dawson, M. (2010). Programming with PYTHON. Cengage Learning
2. Budd, T. A. (2009). Exploring Python. TMH.
3. Cunningham, K. (2013). Python in 24 Hours. Pearson.
4. Payne, J. (2010). Beginning Python. Wiley Publishing.

- **List of Journals/ Periodicals/ Magazines/ Newspapers/etc.:**

The students will have to refer to past issues of the following journals/videos/ebooks in order to get relevant information pertaining to the subject.

- E-Book: Automate the Boring Stuff with Python by Al Sweigart.
- Journal: Python to learn programming: Journal of Physics
- <https://nptel.ac.in/courses/106106145>
- <http://learnpythonthehardway.org/book/>
- <http://www.tutorialspoint.com/python/>
- <https://docs.python.org/3/tutorial/index.html>
- <http://www.learnpython.org/>
- https://7chan.org/pr/src/Beginning_Python_-_Using_Python_2_6_and_Python_3_1.pdf

- **Session Plan:**

Session No.	Topics/Chapters
1.	Practicals based on Variables, Identifiers, Comments, Literals Constants
2.	Practicals based on operators
3.	Practicals based on expressions
4.	Practicals based on Math
5.	Practicals based on Math
6.	Practicals based on String Formatting Methods
7.	Practicals based on Comparisons, Formatting Strings
8.	Practicals based on Comparisons, Formatting Strings
9.	Practicals based on Numbers to String conversion
10.	Practicals based on Conditional Statements
11.	Practicals based on Conditional Statements
12.	Practicals based on While Loop
13.	Practicals based on For Loop
14.	Practicals based on Break and continue
15.	Practicals based on List
16.	Practicals based on List
17.	Practicals based on Tuple and Dictionary
18.	Practicals based on Tuple and Dictionary
19.	Practicals based on Functions & Modules
20.	Practicals based on Functions & Modules
21.	Practicals based on Functions & Modules
22.	Practicals based on Functions & Modules
23.	Practicals based on Functions & Modules
24.	Practicals based on Functions & Modules
25.	Practicals based on Functions & Modules
26.	Practicals based on Functions & Modules
27.	Practicals based on Functions & Modules
28.	Practicals based on Classes, Exceptions, Handling Exceptions
29.	Practicals based on Classes, Errors, Exceptions, Handling Exceptions
30.	Practicals based on Errors, Exceptions, Handling Exceptions
31.	Practicals based on Errors, Exceptions, Handling Exceptions
32.	Practicals based on Errors, Exceptions, Handling Exceptions
33.	Practicals based on Try Finally, The with statement
34.	Practicals based on Try Finally, The with statement
35.	Practicals based on Try Finally, The with statement
36.	Practicals based on concept of scope of variables
37.	Practicals based on GUI using Tkinter
38.	Practicals based on GUI using Tkinter
39.	Practicals based on GUI using Tkinter

40.	Practicals based on GUI using Tkinter
41.	Practicals based on GUI using Tkinter
42.	Practicals based on GUI using Tkinter
43.	Practicals based on GUI using Tkinter
44.	Practicals based on Python Libraries
45.	Practicals based on Python Libraries

GLS UNIVERSITY