

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT205	PROBLEM SOLVING USING PYTHON	PCC	3	1	0	4

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of writing readable PYTHON programs to solve computational problems that they may have to solve in their professional life. The course content is decided to cover the essential programming fundamentals which can be taught within the given slots in the curriculum. This course has got 3 lecture hours and 1 tutorial hour per week for learning and practicing programming using PYTHON. The instructor is supposed to give homework/assignments to write simple programs in the rough record as and when the required theory part is covered in the class. The programs that require time and effort can be done in the Lab sessions. The students are expected to come prepared with the required program written in the rough record for the lab classes.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcomes	Bloom's Category
CO 1	Write programs using Python and learn its execution environment	Understand
CO 2	Apply programs to implement various computational tasks which requires loops and conditional statements	Apply
CO 3	Write programs using functions and packages	Understand
CO 4	Apply programs to implement the concept of file handling using python	Apply
CO 5	Design object oriented programs to implement daily life problems and their solutions	Apply

Mapping of course outcomes with program outcomes

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	1	1	1	-	-	-	-	-	-	-	-	-
CO 2	2	2	1	1	-	-	-	-	-	-	-	-
CO 3	3	2	2	2	2	-	-	-	-	-	-	1
CO 4	1	1	1	2	1	-	-	-	-	-	-	1
CO 5	3	1	3	2	1	1	-	-	-	-	-	1

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	25
Understand	20	20	35
Apply	20	20	40
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Course Level Assessment Questions**Course Outcome 1 (CO1):**

Describe identifiers, variables, keywords, expressions and statements, Operators and operands.

Describe the expression evaluation in Python.

Describe the syntax of control statements in Python.

Write programs to solve problems using various control structures.

Differentiate between Break and Continue.

Course Outcome 2 (CO2):

Build applications of various string manipulations by using methods and functions available with string module in python.

Build applications of various List operations, including matrix representation.
Build applications of Tuple and various Tuple operations.
Build applications of Dictionary and its related operations, functions and methods.

Course Outcome 3(CO3):

Describe function definition and function access in python.
Differentiate between parameters and arguments.
Differentiate between type conversion and coercion.
Describe mathematical & lambda functions in Python.
Explain the concept of composition of functions in Python.
Explain Recursion and its implementation in Python.
Explain the concepts of modules and packages in Python. How and why import method is used.
Write programs to solve problems using the concept of functions and recursion.

Course Outcome 4 (CO4):

Apply the concept of file operations including opening, writing to and reading from files, and writing variables using Python.
Build applications to demonstrate the concept of Pickling.
Build programs to demonstrate the concept of Exception Handling in python.

Course Outcome 5 (CO5):

Build classes using python & Access class variables.
Build programs in Python to demonstrate the use of instances as arguments and return values.
Build programs in Python to demonstrate the concept of Constructors, class attributes and destructors.
Model the concept of Inheritance using Python.

Model Question paper

Course Code: ITT 205

Course Name: Problem Solving Using Python

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Write a python program to find the sum of all odd terms in a group of n numbers entered by the user.
2. What is the use of *pass* statement in Python?
3. Write a Python code to check whether two strings are equal or not.
4. Write a Python code to search an element in a list.
5. List the advantages of using functions in a program.
6. State the use of dump method with suitable example.

7. Write a function exists() which returns True if the given file exists and False if it does not.
8. Why exceptional handling is required in programming?
9. Describe the concept of Constructor with an example.
10. Explain the purpose of __init__() method in Python.

(10x3=30)

Part B**Answer any one Question from each module. Each question carries 14 Marks**

- 11. (a)** Write a Python program to find the largest and second largest of n numbers. Assume $n \geq 3$ and all the numbers are distinct. No sorting algorithm should be used. (7)
- (b)** What are arithmetic operators used in Python? Explain each using an example (7)

OR

- 12. (a)** Write a Python program to print the odd composite numbers between m and n, where m and n are positive integers greater than 1. (8)
- (b)** Define the following
- i) Atoms
 - ii) Identifiers
 - iii) Literals

- 13. (a)** Write a Python code to add two matrices using list. (6)
- (b)** Write a Python program to reverse a string and print whether its palindrome or not. (8)

OR

- 14. (a)** How to create Dictionary in Python? Write a Python program to read and display a sparse matrix using dictionary. (7)
- (b)** Write a program to
- i) convert all small letters in a string into capital letters
 - ii) find the occurrence of a given substring. (7)

- 15.(a)** Compare the built-in functions int() and str() with examples. (4)
- (b)** Write a program using function to display a multiplication table of $n \times n$ size, for any given 'n'. (5)
- (c)** Write a program using function to display a find the binomial coefficient, nC_r . (5)

OR

- 16 (a).** What is recursion? Write a recursive function to find the factorial of a number. (6)
- (b)** Write a program using function to check the type of a triangle by getting the values from the user. (8)
- 17. (a)** How exceptions are handled in Python? Illustrate with example. (10)
- (b)** Write a program to read numbers sorted in one file and store the sorted numbers in another file after deleting duplicates. (4)

OR

18. (a). Describe the use of try-except method in Python with suitable Illustrations. (6)

(b) Write a Python code to read a text file, copy the contents to another file after removing the blank lines. (8)

19. (a). Write a Python code to create a class named 'Member' having the following members: Data members Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same. (8)

(b) Create a class person with attributes Name, age, salary and a method display() for showing the details. Create two instances of the class and call the method for each instance. (6)

OR

20. (a) Define the terms class, attribute, method and instance with the help of an example. (4)

(b) Write a Python program to find out the total mark of a student using multiple inheritance. Declare a Student class. Student class should have the member functions for accept and display student details. Declare a Sports class to accept and display sports marks of the student. Derive a class statement from both the classes. This class should have the member functions to accept marks of three subjects and add those marks with sports marks and display the result. (10)

(14 x 5=70)

Syllabus

Module 1	9 hours
Introduction To Python: Understanding Python-identifiers, variables, keywords, expressions and statements, evaluation of expressions, Operators and operands, operator precedence, indentation. Python Program Flow Control: Decision making- if, if..else, elif. Loops - for, while, for...else, while...else, Control statements using pass, continue, break.	
Module 2	9 hours
Strings and lists – string traversal, string slices and comparison with examples, The string module, character classification. List- List values, accessing elements, list membership, Lists and for loops, List operations, List slices, List deletion, Matrices. Tuples - mutability and tuples, tuple assignment, Tuples as return values, Tuple operations. Dictionaries – operations and methods.	
Module 3	9 hours
Python Functions, Modules and Packages: Function definition, calling functions, parameters and arguments, the return statement, type conversion and coercion, composition of functions, Lambda function, mathematical functions, user-defined functions, Recursion, Modules- Built-in modules, creating modules, import statement. Packages in Python - importing modules from a package.	

Module 4	9 hours
Python Files and exceptions: Python file handling, open, write, read text files, writing variables, Directories in Python, Pickling, Exception Handling.	
Module 5	9 hours
Python Object Oriented Programming: Introduction to classes and objects - class definition, attributes, instances, sameness, instances as arguments and return values. Constructor, class attributes and destructors, Inheritance.	

Text Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers, “How to think like a Computer Scientist-Learning with Python”, Green Tea Press, First edition, 2002.
2. Mark Lutz, ”Learning Python: Powerful Object-Oriented Programming”, O’Reilly Media Inc., 5th, 2013

Reference Books

1. Kenneth A. Lambert, B. L. Juneja, “Fundamentals of Python”, Cengage Learning India Pvt. Ltd., 2015.
2. S.A.Kulkarni, “Problem Solving and PYTHON Programming”, 2nd edition, Yes Dee Publishing Pvt Ltd, 2018
3. Mark Summerfield, ”Programming in Python 3: A Complete Introduction to the Python Language”, Pearson Education, 2nd, 2018
4. Yashavant Kanetkar, Aditya Kanetkar, ”Let Us Python”, BPB Publications, 1st Edition, 2019
5. Allen Downey, “Learning with Python”, Dreamtec Press, 1st Edition, 2015
6. <https://docs.python.org/3/reference/>

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	Introduction To Python:	9 hours
1.1	Understanding Python-identifiers, variables, keywords, expressions and statements.	2
1.2	Evaluation of expressions, Operators and operands, operator precedence, indentation	1

1.3	Python Program Flow Control: Decision making- if, if..else, elif.	2
1.4	Loops - for, while, for...else, while...else	2
1.5	Control statements using pass, continue, break.	2
2	Strings and lists:	9 hours
2.1	String traversal, string slices and comparison with examples	1
2.2	The string module, character classification.	1
2.3	List- List values, accessing elements, list membership, Lists and for loops, List operations, List slices, List deletion	2
2.4	Matrices	1
2.5	Tuples- mutability and tuples, tuple assignment, tuples as return values, Tuple operations.	2
2.6	Dictionaries – operations and methods.	2
3	Python Functions, Modules And Packages:	9 hours
3.1	Function definition, calling functions, parameters and arguments, the return statement.	1
3.2	Type conversion and coercion, composition of functions	1
3.3	Lambda function, mathematical functions	1
3.4	user-defined functions	1
3.5	Recursion	1
3.6	Modules -Built-in modules	1
3.7	Creating modules, import statement.	1
3.8	Packages in Python - importing modules from a package.	2
4	Python Files and exceptions:	9 hours
4.1	Python file handling, open, write, read text files	4
4.2	Writing variables	1
4.3	Directories in Python	1
4.4	Pickling	1
4.5	Exception Handling.	2

5	Python Object Oriented Programming:	9 hours
5.1	Introduce classes and objects	1
5.2	Class definition, attributes, instances, sameness	1
5.3	Instances as arguments and return values.	1
5.4	Constructor	2
5.5	Class attributes and destructors	2
5.6	Inheritance	2

