INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad-500043

COMPUTER SCIENCE AND ENGINEERING

TUTORIAL QUESTION BANK

Course Title	**OBJECT ORIENTED PROGRAMMING THROUGH PYTHON**				
Course Code	AITB01				
Programme	B.Tech				
Semester	III	CSE | IT			
Course Type	Core				
Regulation	**IARE - R18**				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3		3	-	-
Chief Coordinator	Dr.M Purushotham Reddy, Associate Professor				
Course Faculty	Dr.R Obulakonda Reddy, Associate Professor Ms. A Lakshmi, Assistant Professor Ms. M Ashoka Deepthi, Assistant Professor Ms. B Tejaswi, Assistant Professor Mr. P Ravinder, Assistant Professor				

COURSE OBJECTIVES:

The course should enable the students to:	
I	Understand the fundamentals of Python programming concepts and its applications.
II	Understand the object-oriented concepts using Python in problem solving.
III	Apply string handling and function basics to solve real-time problems.
IV	Illustrate the method of solving errors using exception handling.
V	Design and implement programs using multi threading concepts.

COURSE OUTCOMES (COs):

CO 1	Describe Features of Python, Data types, Operators, Input and output, Control Statements, Features of Object oriented programming system, Classes and Objects, Encapsulation, Abstraction, Inheritance, Polymorphism
CO 2	Determine Creating a class, The Self variable, Constructor, Types of Variable, Namespaces, Types of Methods, Inheritance and Polymorphism Constructors in inheritance, The super() method, Types of inheritance, Polymorphism, Abstract classes and Interfaces
CO 3	Understand Creating strings and basic operations on strings, String testing methods, Defining a function, Calling a function, Returning multiple values from a function, Functions are first class objects, Formal and actual arguments, Positional arguments, Recursive functions

CO 4	Explore the concept of Errors in a Python program, Exceptions, Exception handling, Types of exceptions, The Except block, The assert statement, user-defined exceptions
CO 5	Knowledge The Root window, Fonts and colors, Working with containers, Canvas, Frames, Widgets, Button widget, Label Widget, Message widget, Text widget, Radio button Widget, Entry widget

COURSE LEARNING OUTCOMES (CLOs):

AITB01.01	Describe the Features of Python, Data types.
AITB01.02	Summarize the concept of Operators, Input and output, Control Statements.
AITB01.03	Identify the features of Object Oriented Programming System (OOPS).
AITB01.04	Use the concept of Classes and Objects, Encapsulation.
AITB01.05	Describe Abstraction, Inheritance, and Polymorphism.
AITB01.06	Determine Creating a class, The Self variable.
AITB01.07	Understand types of variable, Namespaces.
AITB01.08	Determine types of Methods, Inheritance and Polymorphism.
AITB01.09	Use Constructors in inheritance, the super() method.
AITB01.10	Illustrate types of inheritance, Polymorphism, Abstract classes and Interfaces.
AITB01.11	Understand Creating strings and basic operations on strings.
AITB01.12	Analyze the concept of String testing methods, Defining a function.
AITB01.13	Illustrate Calling a function, Returning multiple values from a function.
AITB01.14	Contrast the Usage of Functions are first class objects, Formal and actual arguments,
AITB01.15	Define Positional arguments, Recursive functions.
AITB01.16	Discuss the concept of Errors in a Python program.
AITB01.17	Understand Exceptions, Exception handling.
AITB01.18	Summarize the concept of types of exceptions.
AITB01.19	Discuss the Except block, the assert statement.
AITB01.20	Understand the concept of user-defined exceptions.
AITB01.21	Knowledge about the Root window, Fonts and colors.
AITB01.22	Apply Working with containers, Canvas.
AITB01.23	Understand Widgets, Button widget, Label Widget.
AITB01.24	Implement Message widget, Text widget.
AITB01.25	Illustrate Radio button Widget, Entry widget.

TUTORIAL QUESTION BANK

MODULE- I				
INTRODUCTION TO PYTHON AND OBJECT ORIENTED CONCEPTS				
Part - A (Short Answer Questions)				
S No	**QUESTIONS**	**Blooms Taxonomy Level**	**Course Outcomes**	**Course Learning Outcomes (CLOs)**
1	What are the features of Python programming language?	Remember	CO 1	AITB01.01
2	What is the role of Python Interactive shell?	Understand	CO 1	AITB01.01
3	What are the different modes of working in Python?	Remember	CO 1	AITB01.01
4	What are the rules for identifier?	Remember	CO 1	AITB01.01
5	How to check the number of keywords in Python?	Understand	CO 1	AITB01.03
6	What are the standard data types in Python?	Remember	CO 1	AITB01.01
7	Define a tuple.	Understand	CO 1	AITB01.01
8	Define a List.	Remember	CO 1	AITB01.01
9	Define a Set and its types.	Remember	CO 1	AITB01.01
10	Define a dictionary.	Understand	CO 1	AITB01.01
11	List out the operators in Python?	Understand	CO 1	AITB01.02
12	Define a control structure?	Remember	CO 1	AITB01.02
13	What are the various types of loops in Python?	Understand	CO 1	AITB01.02
14	Define a class.	Understand	CO 1	AITB01.03
15	Define an object.	Remember	CO 1	AITB01.03
16	Define a method.	Understand	CO 1	AITB01.03
17	List out the features of object oriented programming.	Understand	CO 1	AITB01.03
18	Define Encapsulation.	Remember	CO 1	AITB01.04
19	Define Inheritance.	Understand	CO 1	AITB01.05
20	Define Abstraction.	Remember	CO 1	AITB01.05
Part - B (Long Answer Questions)				
1	Explain the features of Python programming language in detail.	Understand	CO 1	AITB01.01
2	What is an operator and explain about the arithmetic operators and assignment operators in Python with example.	Understand	CO 1	AITB01.02
3	Describe about input statements in Python and formatting strings with examples.	Remember	CO 1	AITB01.02
4	Explain about features of Object Oriented Programming compared with the procedure oriented programming.	Understand	CO 1	AITB01.03
5	Explain in detail about the if statement and if-else statement with examples.	Understand	CO 1	AITB01.02
6	Explain the concept of classes and objects in detail with any real time example.	Understand	CO 1	AITB01.04
7	Illustrate the if-elif-else statement and while loop with examples.	Understand	CO 1	AITB01.02
8	Explain about built-in data types and sequences in Python with examples.	Remember	CO 1	AITB01.01
9	Describe the set data type in Python and operations on set data types.	Understand	CO 1	AITB01.01
10	Explain about literals in Python and types of literals in Python with example.	Understand	CO 1	AITB01.01
11	Explain about encapsulation in Object Oriented Programming with example.	Remember	CO 1	AITB01.04
12	Describe about output statements in Python and formatting strings with examples.	Understand	CO 1	AITB01.02
13	Explain about abstraction in Object Oriented Programming with example.	Understand	CO 1	AITB01.05
14	Explain about user defined data types and constants in Python in detail.	Remember	CO 1	AITB01.01
15	Explain about inheritance in Object Oriented Programming with example.	Understand	CO 1	AITB01.05
16	Describe the logical operators and Boolean operators with example.	Remember	CO 1	AITB01.02
17	Explain about the unary operators and relational operators in Python with example.	Understand	CO 1	AITB01.02
18	Explain about Bitwise operators and membership operators in Python with example.	Understand	CO 1	AITB01.02
19	Describe the for loop and the break statement and the continue statement in Python with examples.	Understand	CO 1	AITB01.02
20	Explain about identity operators and operator precedence and associativity with example.	Understand	CO 1	AITB01.02

	Part - C (Problem Solving and Critical Thinking Questions)			
1	Write a Python program to create all possible strings by using 'a', 'e', 'i', 'o', 'u'. Use the characters exactly once.	Understand	CO 1	AITB01.01
2	Write code snippets in Python to perform the following a. Accessing elements of a tuple b. Modifying elements of a tuple c. Deleting elements of a tuple	Understand	CO 1	AITB01.01
3	Write a Python program to count the number of words in a text file	Understand	CO 1	AITB01.02
4	Write a Python program using while loop first N numbers divisible by 5.	Understand	CO 1	AITB01.02
5	Write a simple program in Python to convert decimal number into binary, octal and hexadecimal number system in Python.	Understand	CO 1	AITB01.01

6	What is output of following code −
	class Count:
	def __init__(self, count=0):
	self__count=count
	a=Count(2)
	b=Count(2)
	print(id(a)==id(b), end = " ")
	
	c= "hello"
	d= "hello"
	print(id(c)==id(d))
		Understand	CO 1	AITB01.02

7	Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.	Understand	CO 1	AITB01.01
8	Write a Python program to construct the following pattern, using a nested for loop. *	Understand	CO 1	AITB01.01
9	Write a Python program to add two positive integers without using the '+' operator.	Understand	CO 1	AITB01.02
10	Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.	Understand	CO 1	AITB01.02
	MODULE-II			
	PYTHON CLASSES AND OBJECTS			
	Part – A (Short Answer Questions)			
1	Define Class and Object.	Remember	CO 2	AITB01.06
2	Explain how the class is defined, object is created, and methods are invoked in Python.	Remember	CO 2	AITB01.06
3	Discuss the use of init method in Python.	Remember	CO 2	AITB01.06
4	Why Objects are mutable?	Understand	CO 2	AITB01.06
5	List the features of the object oriented programming through Python.	Understand	CO 2	AITB01.06
6	What is inheritance?	Remember	CO 2	AITB01.08
7	List different types of inheritance.	Understand	CO 2	AITB01.08
8	What is namespace in Python?	Understand	CO 2	AITB01.07
9	What is self in Python?	Understand	CO 2	AITB01.07
10	How are classes created in Python?	Remember	CO 2	AITB01.06
11	What is Polymorphism in Python?	Understand	CO 2	AITB01.10
12	What is multiple inheritance?	Remember	CO 2	AITB01.08
13	What is operator overloading?	Understand	CO 2	AITB01.08
14	What is meant single inheritance ?	Remember	CO 2	AITB01.08
15	What is the purpose of inheritance in object oriented program in Python?	Understand	CO 2	AITB01.08
16	What does the super() do in Python?	Understand	CO 2	AITB01.09

17	What is __init__ in Python?	Remember	CO 2	AITB01.07
18	What is the difference between abstract class and interface?	Remember	CO 2	AITB01.10
19	What is abstract method in Python ?	Understand	CO 2	AITB01.10
20	Define multilevel inheritance?	Remember	CO 2	AITB01.08
	Part - B (Long Answer Questions)			
1	What is polymorphism? Explain the polymorphism with suitable example program.	Understand	CO 2	AITB01.10
2	What is inheritance? Explain with example and write a program for representing inheritance.	Understand	CO 2	AITB01.08
3	List different types of inheritance and Explain each and every one with suitable examples.	Understand	CO 2	AITB01.08
4	Defining the following with examples. i. Creating a class ii. Constructor iii. The self variable	Remember	CO 2	AITB01.06
5	Explain in detail about class, objects and methods with suitable examples?	Understand	CO 2	AITB01.06
6	What is a namespace? How do you resolve the name conflicts using namespaces? Explain with an example.	Understand	CO 2	AITB01.07
7	Explain the super( ) method with two suitable examples.	Understand	CO 2	AITB01.09
8	What is the relationship between a class and an object? Explain this with two suitable examples.	Remember	CO 2	AITB01.06
9	What is abstract class? Explain abstract class method with example.	Understand	CO 2	AITB01.06
10	Why does the object-oriented philosophy need functions to be defined inside the classes? What could be the advantage?	Understand	CO 2	AITB01.06
11	List different methods of realizing polymorphism and explain them with example.	Remember	CO 2	AITB01.10
12	Explain multiple views of an object with suitable example.	Understand	CO 2	AITB01.06
13	Define class. Explain Nested classes and local classes with an example.	Understand	CO 2	AITB01.06
14	Explain differences between various types of inheritance?	Remember	CO 2	AITB01.08
15	What is a class? What is the relation between an object and a class? Write a program which shows how to define a class, how to access member functions and how to create and access objects in Python.	Remember	CO 2	AITB01.06
16	What is inheritance? Explain with example how to inherit a class in Python.	Understand	CO 2	AITB01.06
17	What is a nested class? What are its advantages? How it is defined and declared in Python?	Understand	CO 2	AITB01.06
18	Define inheritance and list different types of inheritance. How multilevel inheritance is different from multiple inheritance?	Remember	CO 2	AITB01.08
19	Define abstract class? Write differences between abstract classes and interfaces with examples.	Remember	CO 2	AITB01.10
20	Explain the following with examples. i. Polymorphism ii. Inheritance iii. Abstract class	Understand	CO 2	AITB01.10
	Part - C (Problem Solving and Critical Thinking Questions)			
1	Create a class whose object represents a complex number (A complex number contains a real part and an imaginary part). Write a program so that it is possible to add two objects of this class and store the result in third object.	Understand	CO 2	AITB01.06
2	Explain public, private and protected access specifiers and show the ambiguity in multiple and multilevel inheritance.	Understand	CO 2	AITB01.08
3	Create a class called Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0. and another should initialize it to fixed values. A member function should display it, in 11:59:59 format. Write a program to add time of two objects by overloading '+' operator.	Remember	CO 2	AITB01.06
5	Explain the inheritance. List different types of inheritance. Write differences between them.	Understand	CO 2	AITB01.08
6	Justify “Class is a template while Object is data”.	Understand	CO 2	AITB01.06
7	Describe polymorphism as applied to OOP. Explain polymorphism with examples.	Understand	CO 2	AITB01.10

8	Describe abstract classes and interfaces. Explain differences between abstract class and interface.	Understand	CO 2	AITB01.10
9	What are different forms of inheritance? Give an example for each and every inheritance.	Understand	CO 2	AITB01.08
10	Explain how base class member functions can be invoked in a derived class if the derived class also has a member function with the same name.	Remember	CO 2	AITB01.06
	MODULE -III			
	STRINGS AND FUNCTIONS			
	Part - A (Short Answer Questions)			
1	Define string. Write the syntax of creating a string with example	Remember	CO 3	AITB01.11
2	"There is no difference between single quotes and double quotes while creating the string". Justify the statement.	Remember	CO 3	AITB01.11
3	List different string operations. Write example programs for any three string operations.	Remember	CO 3	AITB01.11
4	List the escape characters that can be used in strings.	Remember	CO 3	AITB01.11
5	Define length of string and what is the predefined function used to find length of string? Illustrate with an example.	Remember	CO 3	AITB01.11
6	Write about indexing concept in strings.	Understand	CO 3	AITB01.11
7	Explain the methods that are used to find substrings in main string?	Understand	CO 3	AITB01.11
8	Write about the following operations on strings i)Slicing ii)Repeating	Understand	CO 3	AITB01.11
9	Explain how to remove spaces from a string. Write related examples	Understand	CO 3	AITB01.11
10	Mention and explain different sting testing methods.	Remember	CO 3	AITB01.12
	CIE-II			
11	Define a function. Write the syntax of defining a function with example	Remember	CO 3	AITB01.12
12	Specify the process of calling a function.	Understand	CO 3	AITB01.12
13	Write the difference between functions returning single value and functions returning multiple values.	Understand	CO 3	AITB01.13
14	Compare actual and formal arguments with example.	Understand	CO 3	AITB01.14
15	Mention different types of arguments. Define positional arguments	Remember	CO 3	AITB01.15
16	List the advantages of functions?	Remember	CO 3	AITB01.12
17	Write the difference between a function and method.	Understand	CO 3	AITB01.12
18	Why functions in Python are called as first class objects?	Understand	CO 3	AITB01.14
19	Write a Python function that accepts two values and finds their sum.	Remember	CO 3	AITB01.12
20	Define recursive function and illustrate with example program.	Remember	CO 3	AITB01.15
	Part – B (Long Answer Questions)			
1	Summarize the escape characters that can be used in strings with an example	Understand	CO 3	AITB01.11
2	Discuss the following operations on strings i)Length of string ii)Indexing in strings iii)counting substrings in a string	Understand	CO 3	AITB01.11
3	Explain the following methods i)upper() ii)lower() iii)swapcase() iv)title()	Understand	CO 3	AITB01.12
4	Explain different string and character testing methods with examples	Understand	CO 3	AITB01.11
5	Explain how can we split and join strings in Python with an example	Understand	CO 3	AITB01.11
6	Write a Python program to display all positions of a substring in a given main string.	Remember	CO 3	AITB01.11
7	Illustrate the concept of slicing the strings with an example program.	Remember	CO 3	AITB01.11
8	Discuss about the following methods that are used to remove spaces from a string. i)rstrip() ii)lstrip() iii)strip()	Understand	CO 3	AITB01.11
9	Explain the methods that are useful to locate sub strings in a string with example programs.	Understand	CO 3	AITB01.11

10	Write various ways of assigning a group of characters to a variable.	Remember	CO 3	AITB01.11
CIE-II				
11	Explain the following i)Defining a function ii)Calling a function	Understand	CO 3	AITB01.13
12	Explain how functions can return results with an example.	Understand	CO 3	AITB01.13
13	Describe the role of Python interpreter in functions. Explain possible ways of assigning a function.	Understand	CO 3	AITB01.14
14	Draw and explain the steps involved in Towers of Hanoi problem through recursion.	Remember	CO 3	AITB01.15
15	Explain how a function can return multiple values with an example.	Understand	CO 3	AITB01.13
16	Discuss about i)Positional arguments ii)Variable length arguments	Remember	CO 3	AITB01.15
17	Write a Python program to implement Towers of Hanoi problem using recursion.	Remember	CO 3	AITB01.15
18	List and explain different ways of passing values to function with examples.	Understand	CO 3	AITB01.14
19	Write a Python function to check the given number is prime or not.	Remember	CO 3	AITB01.13
20	Write a Python function to check the given number is palindrome or not.	Remember	CO 3	AITB01.13
Part – C (Problem Solving and Critical Thinking)				
1	Write a Python program to access characters of a string using for loop.	Remember	CO 3	AITB01.11
2	Write a Python program that implements i)string concatenation ii)string comparison iii)string length	Understand	CO 3	AITB01.11
3	Write a Python program to find the first occurrence of sub string in given main string.	Understand	CO 3	AITB01.11
4	Write Python program that implements different string testing methods	Understand	CO 3	AITB01.12
5	Write a Python program to update or delete a string	Remember	CO 3	AITB01.11
CIE-II				
06	Write a Python function i)to test whether a number is even or odd. ii)to calculate factorial value of numbers from 1 to 10	Understand	CO 3	AITB01.12
07	Write a Python program to understand the positional arguments of a function.	Remember	CO 3	AITB01.15

08	Predict the output of following code
	def swap(x, y):
	temp = x;
	x = y;
	y = temp;
	# Driver code
	x = 2
	y = 3
	swap(x, y)
	print(x)
	print(y)
		Understand	CO 3	AITB01.13

09	Write a Python function to sum all the numbers in a list. Sample List : (8, 2, 3, 0, 7) Expected Output : 20	Understand	CO 3	AITB01.13
10	Write a Python program to calculate factorial of a given number using recursion concept.	Understand	CO 3	AITB01.15
MODULE –IV				
EXCEPTION HANDLING				
Part – A (Short Answer Questions)				
1	How many except statements can a try-except block have?	Remember	CO 4	AITB01.17
2	What is an exception?	Remember	CO 4	AITB01.17
3	When will the else part of try-except-else be executed?	Remember	CO 4	AITB01.18
4	Can we write only try block without catch and finally blocks?	Understand	CO 4	AITB01.19
5	Can we keep other statements in between try, catch and finally blocks?	Remember	CO 4	AITB01.19
6	How can you handle an exception?	Understand	CO 4	AITB01.17

7	How can you catch multiple exceptions?	Understand	CO 4	AITB01.18
8	What is try-except?	Remember	CO 4	AITB01.18
9	What is try-finally statement?	Understand	CO 4	AITB01.18
10	Explain raise syntax.	Remember	CO 4	AITB01.19
11	How to handle exceptions with try-except	Remember	CO 4	AITB01.19
12	How to handle all types of exception with except?	Understand	CO 4	AITB01.18
13	How to handle multiple exceptions with except?	Understand	CO 4	AITB01.18
14	How to handle exceptions with try-finally?	Understand	CO 4	AITB01.18
15	How to raise exception with arguments?	Understand	CO 4	AITB01.17
16	How to create custom exceptions in Python?	Understand	CO 4	AITB01.18
17	Identify the type of error in the codes shown below. Print("Good Morning") print("Good night)	Remember	CO 4	AITB01.20
18	Is the following code valid? # Do something except: # Do something else: # Do something	Remember	CO 4	AITB01.20
19	Which of the following is not an exception handling keyword in Python? a) try b) except c) accept d) finally	Understand	CO 4	AITB01.19
20	What is the output of the following code? Def foo(): try: return 1 finally: return 2 k = foo() print(k)	Remember	CO 4	AITB01.18
21	What is the output of the following code? Def foo(): try: print(1) finally: print(2) foo()	Understand	CO 4	AITB01.19

22	What is the output of the following?
	Try:
	if '1' != 1:
	raise "someError"
	else:
	print("someError has not occurred")
	except "someError":
	print ("someError has occurred")
		Understand	CO 4	AITB01.19

23	What is the output of the code shown below? X=10 y=8 assert x>y, 'X too small'	Remember	CO 4	AITB01.17
24	What is the output of the code shown below? #generator def f(x): yield x+1 g=f(8) print(next(g))	Understand	CO 4	AITB01.17
25	What is the output of the code shown below? Def f(x): yield x+1	Remember	CO 4	AITB01.18

	print(“test”) yield x+2 g=f(9)			
26	What is the output of the code shown below? Def f(x): yield x+1 print(“test”) yield x+2 g=f(10) print(next(g)) print(next(g))	Understand	CO 4	AITB01.18
PART – B (LONG ANSWER QUESTIONS)				
1	How the exceptions are handled in Python? Explain exception handling mechanism in Python?	Understand	CO 4	AITB01.18
2	What is the difference between error and exception in Python?	Understand	CO 4	AITB01.18
3	Can we keep other statements in between try, catch and finally blocks? Explain	Understand	CO 4	AITB01.19
4	What is unreachable catch block error?	Understand	CO 4	AITB01.19
5	Explain the hierarchy of exceptions in Python?	Remember	CO 4	AITB01.17
6	What are run time exceptions in Python? Give example?	Remember	CO 4	AITB01.18
7	Can we keep the statements after finally block If the control is returning from the finally block itself?	Understand	CO 4	AITB01.19
8	Does finally block get executed If either try or catch blocks are returning the control?	Remember	CO 4	AITB01.19
9	Can we throw an exception manually? If yes, how?	Remember	CO 4	AITB01.17
10	What are the legal combinations of try, catch and finally blocks? Explain?	Remember	CO 4	AITB01.19
11	Can we keep other statements in between try, catch and finally blocks? Explain	Understand	CO 4	AITB01.19
12	What is unreachable catch block error?	Remember	CO 4	AITB01.16
13	How do you create customized exceptions in Python?	Understand	CO 4	AITB01.18
14	Can one block of except statements handle multiple exceptions? Explain in Detail?	Understand	CO 4	AITB01.18
15	How can you catch multiple exceptions?	Remember	CO 4	AITB01.18
16	What are assertions? Explain about the assertions.	Understand	CO 4	AITB01.19
17	What is the difference between an exception and error? Explain with program?	Remember	CO 4	AITB01.17
18	What are the rules in Python we need to follow when overriding a method that throws an exception?	Understand	CO 4	AITB01.16
19	How to handle exceptions with try-finally?	Remember	CO 4	AITB01.18
PART – C (PROBLEM SOLVING AND CRITICAL THINKING QUESTIONS)				

1	What happens if the file is not found in the code shown below?
	A=False
	while not a:
	try:
	f_n = input(“Enter file name”)
	i_f = open(f_n, ‘r’)
	except:
	print(“Input file not found”)
		Understand	CO 4	AITB01.17

2	What is the output of the code shown below if the input entered is 6?
	valid = False
	while not valid:
	try:
	n=int(input("Enter a number"))
	while n%2==0:
	print("Bye")
	valid = True
	except ValueError:
	print("Invalid")
		Understand	CO 4	AITB01.18

3	Let’s take example in which trying to open a file in the READ mode. Then perform a WRITE operation on it. Upon execution, it’ll throw an exception. try: fob = open("test", "r") fob.write("It's my test file to verify exception handling in Python!!")	Understand	CO 4	AITB01.17

	except IOError: print "Error: can\\'t find the file or read data" else: print "Write operation is performed successfully on the file" What is the output the above code produces?			
4	Justify that we can either define an "**except**" or a "**finally**" clause with every try block. You can't club these together. Also, you shouldn't use the "**else**" clause along with a "**finally**" clause.	Understand	CO 4	AITB01.17
5	Compare the two codes shown below and state the output if the input entered in each case is -6? CODE 1 import math num=int(input("Enter a number of whose factorial you want to find")) print(math.factorial(num)) CODE 2 num=int(input("Enter a number of whose factorial you want to find")) print(math.factorial(num))	Understand	CO 4	AITB01.16

6	What is the output of the following code?
	def a():
	try:
	f(x, 4)
	finally:
	print('after f')
	print('after f?')
	a()
		Understand	CO 4	AITB01.18

7	What is the output of the code shown below?
	def getMonth(m):
	if m<1 or m>12:
	raise ValueError("Invalid")
	print(m)
	getMonth(6)
		Understand	CO 4	AITB01.16

8	A try statement can have more than one except clause, to specify handlers for different exceptions. Explain with example program.	Understand	CO 4	AITB01.17
9	In Python, you can use **else clause** on try-except block which must be present after all the except clauses. The code enters the else block only if the try clause does not raise an exception. Justify the above statement?	Understand	CO 4	AITB01.18
10	In Python Reraising the exception, that has been caught in the **except** block. Explain in detail with a program?	Understand	CO 4	AITB01.17
	MODULE -V			
	GRAPHICAL USER INTERFACE			
	Part - A (Short Answer Questions)			
1	Define root window.	Remember	CO 5	AITB01.21
2	What are fonts and colors? Explain.	Understand	CO 5	AITB01.22
3	Define containers.	Remember	CO 5	AITB01.22
4	Define Canvas.	Remember	CO 5	AITB01.23
5	Write the types Widgets.	Remember	CO 5	AITB01.23
6	Define frames.	Remember	CO 5	AITB01.24
7	Define button widget.	Remember	CO 5	AITB01.24
8	Write label widget.	Remember	CO 5	AITB01.24
9	Write message widget.	Remember	CO 5	AITB01.25
10	Define radio button Widget.	Remember	CO 5	AITB01.25
11	Define entry widget.	Remember	CO 5	AITB01.23
	Part - B (Long Answer Questions)			
1	Demonstrate and write types of widgets.	Understand	CO 5	AITB01.23
2	Write the working procedure of containers.	Remember	CO 5	AITB01.22
3	Write the Python code for canvas and frames.	Understand	CO 5	AITB01.22
4	How to create a button widget in Python?	Understand	CO 5	AITB01.23
5	Write the Python code for label Widget.	Remember	CO 5	AITB01.23
6	Distinguish message widget and text widget.	Understand	CO 5	AITB01.24

7	How to create message widget by using Python?	Understand	CO 5	AITB01.24
8	Write the Python code for text widget.	Remember	CO 5	AITB01.24
9	How to create radio button widget?	Remember	CO 5	AITB01.25
10	Write the Python code for entry widget.	Remember	CO 5	AITB01.25
Part – C (Problem Solving and Critical Thinking)				
1	Demonstrate form application from the experimental machine learning to interactive with data mining exploration using Python	Understand	CO 5	AITB01.22
2	What is Python widget? Explain interactive linear and non linear regression model	Understand	CO 5	AITB01.23
3	What exactly are "containers" in Python? what are all the Python container types?	Understand	CO 5	AITB01.23
4	How do you create a GUI in Python? Is Python good for desktop application?	Understand	CO 5	AITB01.23
5	Consider a Python GUI program that produces a window with the following widgets 1. A text box to display the value of one element of a given list 2. A button to retrieve the previous value in that list(if there is one).This button is displayed if there is no previous value in the list	Remember	CO 5	AITB01.23
6	Consider a Python GUI program that produces a window with the following widgets 1. A button to retrieve the next value in that list(if there is one).This button is displayed if there is no next value in the list 2. A label to display the number of the item being displayed and the total number of items ("Example 1/5").	Remember	CO 5	AITB01.23

Prepared by:
Dr.M Purushotham Reddy, Associate Professor

HOD, IT