

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST - 2 EXAMINATION - 2024

M.Tech - I Semester (ECE)

COURSE CODE (CREDITS): 21M11EC113

MAX. MARKS: 25

COURSE NAME: Object Oriented Programming

COURSE INSTRUCTORS: Dr. Naveen Jaglan

MAX. TIME: 1 Hour 30 Minutes

*Note: (a) All questions are compulsory.*

*(b) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems.*

Q.No	Question	CO	Marks
Q1	Write a single program to: (a) Create a Rectangle class in Python language, allowing you to build a rectangle with length and width attributes, (b) Create a Perimeter() method to calculate the perimeter of the rectangle and a Area() method to calculate the area of the rectangle, (c) Create a method display() that display the length, width, perimeter and area of an object created using an instantiation on rectangle class, (d) Create a Parallelepiped child class inheriting from the Rectangle class and with a height attribute and another Volume() method to calculate the volume of the Parallelepiped.	CO-4	5
Q2	Write a generator function that produces a sequence of Fibonacci numbers. What is the purpose of the StopIteration exception?	CO-3	4
Q3	Write a Python program to count total number of uppercase and lowercase characters in a text file. Count total number of lines and count the total number of lines starting with 'A', 'B', and 'C' in the same text file.	CO-2	3
Q4	What is the purpose of the try...except...else...raise...finally block in python exception handling? Write a python program that executes division and handles an ArithmeticError exception if there is an arithmetic error.	CO-3	3
Q5	With the help of a suitable program explain the class methods, static methods and Instance methods.	CO-3	4
Q6	Find the outputs of following python programmes:	CO-4	6

(a)

```
class A:
    def __init__(self):
        self.multiply(15)
    def multiply(self, i):
        self.i = 4 * i;
class B(A):
    def __init__(self):
        super().__init__()
        print(self.i)

    def multiply(self, i):
        self.i = 2 * i;
obj = B()
```

(b)

```
class A:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __str__(self):
        return 1
    def __eq__(self, other):
        return self.x * self.y == other.x * other.y
obj1 = A(5, 2)
obj2 = A(2, 5)
print(obj1 == obj2)
```

(c)

```
def f(x):
    def f1(a, b):
        print("hello")
        if b==0:
            print("NO")
            return
        return f(a, b)
    return f1
@f
def f(a, b):
    return a%b
f(4,0)
```

(d)

```
class A:
    def test(self):
        print("test of A called")
class B(A):
    def test(self):
        print("test of B called")
        super().test()
class C(A):
    def test(self):
        print("test of C called")
        super().test()
class D(B,C):
    def test2(self):
        print("test of D called")
obj=D()
obj.test()
```

(e)

```
l=[1, 0, 2, 0, 'hello', '', []]
list(filter(bool, l))
```

(f)

```
def f(x):
    for i in range(5):
        yield i
g=f(8)
print(list(g))
```