

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST - 3 EXAMINATION - 2024

M.Tech - I Semester (ECE)

COURSE CODE (CREDITS): 21M11EC113 (3)

MAX. MARKS: 35

COURSE NAME: Object Oriented Programming

COURSE INSTRUCTORS: Dr. Naveen Jaglan

MAX. TIME: 2 Hours

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	Can abstract classes be utilized to define a blueprint for classes that interact with hardware or external systems? Provide a practical example using a Python program.	CO-4	4
Q2	Explain the concept of an iterator and how it is implemented in Python. Provide an example of how to create and use an iterator. Describe the purpose of the <code>__iter__()</code> method in an iterable object. How is it different from the <code>next()</code> method?	CO-1	4
Q3	Write a Python generator function that reads a large file line by line. The function should yield each line from the file one at a time, making it memory efficient.	CO-2	5
Q4	Write a Python program to extract year, month and date from an URL using regular expressions: URL= https://www.washingtonpost.com/news/football-insider/wp/2016/09/02/odell-beckhams-fame-rests-on-one-stupid-little-ball-josh-norman-tells-author/	CO-3	4
Q5	In Python, what is the Method Resolution Order (MRO), and how does it determine the order in which methods are called in a class hierarchy? Explain with a suitable program.	CO-3	3
Q6	What are the advantages of using Duck Typing in Python? How does it promote flexibility and dynamic behavior? Explain with the help of a program how can you check if an object can perform a specific method using Duck Typing?	CO-4	4
Q7	Explain the concept of method overriding in Python. How is it different from method overloading? Demonstrate method overriding in Python with an example.	CO-5	3
Q8	Create a parent class Animal with a class variable <code>species_count</code> that tracks how many species have been created. Then, create a child class Dog that inherits from Animal and calls a <code>@classmethod</code> from the parent class to increment the <code>species_count</code> when a new Dog instance is created.	CO-4	5
Q9	What will be the output of the following Python codes?	CO-6	3

(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))
```