MODULE – 4 (ADVANCE PYTHON PROGRAMMING)

Vadhiya Niraj

1. Explain Exception handling? What is an Error in Python?

Ans: - **Explanation:**
Exception handling is used to handle runtime errors, ensuring the program continues running even when an error occurs.

Error in Python:
An error occurs when Python encounters an issue it cannot handle. This raises exceptions.

2. How many except statements can a try-except block have? Name Some built-in exception classes:

Ans: - A try-except block can have multiple except statements, one for each exception type

```
try:
    # Code that may raise exceptions
    pass
except ValueError:
    pass
except IndexError:
    pass
```

3. When will the else part of try-except-else be executed?

Ans: - The else block runs if no exception is raised in the try block

```
try:
    x = 1 / 1
except ZeroDivisionError:
    print("Division by zero error.")
else:
    print("No error occurred.")
```

4. Can one block of except statements handle multiple exception?

Ans: - Yes, you can handle multiple exceptions using a tuple

```
try:
    # Code that may raise exceptions
    pass
except (ValueError, TypeError) as e:
```

```
print(f"Error: {e}")
```

5. When is the finally block executed?

Ans: - The finally block is always executed, regardless of whether an exception occurs or not

```
try:
    # Code that may raise exceptions
    pass
except Exception:
    pass
finally:
    print("This block is always executed.")
```

6. What happens when „1"== 1 is executed?

Ans: - This raises a TypeError because you cannot compare different data types (string vs integer).

7. How Do You Handle Exceptions With Try/Except/Finally In Python? Explain with coding snippets.

Ans: -
```
try:
    num = int(input("Enter a number: "))
except ValueError:
    print("Invalid input! Not a number.")
finally:
    print("This block is always executed.")
```

8. Write python program that user to enter only odd numbers, else will raise an exception.

Ans: -
```
number = int(input("Enter an odd number: "))
if number % 2 == 0:
    raise ValueError("The number is not odd.")
else:
    print(f"Entered odd number: {number}")
```

9. What are oops concepts? Is multiple inheritance supported in java

Ans: - OOP Concepts:

- Encapsulation: Hiding the internal details of an object and exposing only the necessary parts.
- Inheritance: A class can inherit the properties and methods of another class.
- Polymorphism: The ability to use a single method or operator in multiple ways.
- Abstraction: Hiding complex implementation and showing only the essential features.

Multiple Inheritance in Java:
Java does not support multiple inheritance directly with classes, but it can be done through interfaces.

10. How to Define a Class in Python? What Is Self? Give An Example Of A Python Class

Ans: -

```
class MyClass:
    def __init__(self, value):
        self.value = value

    def display(self):
        print(self.value)

obj = MyClass(10)
obj.display()
```

(self) is used to refer to the current instance of the class. It allows access to the class's attributes and methods.

11. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle

Ans: -

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
```

```
    def area(self):
        return self.length * self.width

rect = Rectangle(5, 10)
print("Area:", rect.area())
```

12. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

Ans: - import math

```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return math.pi * self.radius ** 2
```

```
    def perimeter(self):
        return 2 * math.pi * self.radius

circle = Circle(7)
print("Area:", circle.area())
print("Perimeter:", circle.perimeter())
```

13. Explain Inheritance in Python with an example? What is init? Or What Is A Constructor In Python?

Ans: -

```
class Animal:
    def speak(self):
        print("Animal speaks")

class Dog(Animal):
    def speak(self):
        print("Dog barks")
```

```
dog = Dog()
dog.speak()  # Output: Dog barks
```

__init__ (Constructor):
The __init__ method is called when a new object is created, and it initializes the object's state.

14. What is Instantiation in terms of OOP terminology?

Ans: - Instantiation refers to creating an object from a class.

15. What is used to check whether an object o is an instance of class A?

Ans: - isinstance(obj, MyClass)

16. What relationship is appropriate for Course and Faculty?

Ans: - A one-to-many relationship is appropriate, where a faculty member can teach many courses.

17. What relationship is appropriate for Student and Person?

Ans: - A student is a type of person, so the relationship is inheritance.