Module – 4 (Advance python programming)

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

→ Python allows a user to manage errors and exceptional conditions that occur during Program execution.Instead of getting the program crashed when an error occurs exception handling provides a way to catch and handle this error gracefully.

→ An error is an issue in a program that prevents the program from completing its task.

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

→ry-except" block can have multiple except statements. Python will match the first 'except' block that matches it's type.

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

→If no exception are raised after try block then only else part of the try-except-else will be executed.

16. Can one block of except statements handle multiple exceptions?

→ Yes one block of except statement can handle multiple exceptions user just need to write those exceptions in a tuple.

Try:

Pass

Except(exception1,exception2,exception3) as e:

Pas

17. When is the finally block executed?

→Finally block executes every single time the code runs.

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

→Python throws an error as there is == operator with different types (string) and (integer) operands.

21.What are oops concepts? Is multiple inheritance supported in python

→Class

Object

Inheritance

Encapsulation

Abstraction

Polymorphism

multiple Inheritance is supported in python

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

→Class is a collection of data members and member functions.

→“Self” is a convention used in Python classes to refer to the instance of the class. It is the first parameter of methods in a class and is used to access attributes and methods on the instance.

```
    class Rectangle:

def __init__(self,length,width):

    self.length=length

    self.width=width

    def area(self):

    return self.length*self.width

rect=Rectangle(10,5)

    print(f"The area of rectangle is {rect.area()}")
```

26. Explain Inheritance in Python with an example? What is in it? Or What· Is A Constructor In Python?

→ Object of one class can acquire the properties of object of anther class is inheritance

```
class Vehicle:
def __init__(self, brand, model):
self.brand = brand
self.model = model
def display_info(self):
return f"Brand: {self.brand}, Model: {self.model}"
    class Car(Vehicle):
def __init__(self, brand, model, number_of_doors):
super().__init__(brand, model)
    self.number_of_doors = number_of_doors
def car_info(self):
return f"This car has {self.number_of_doors} doors."
    my_car = Car(brand="Toyota", model="Corolla", number_of_doors=4)
```

There are 5 types of inheritance:

1)Single Inheritance

2)Multilevel Inheritance

3)Multiple Inheritance

4)Hierarchical Inheritance

5)Hybrid Inheritance

27.What is Instantiation in terms of OOP terminology

→ instantiation is the process of creating an instance of a class, or object, from a blueprint.

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

→ built-in function isinstance() to check whether an object is an instance of a specified class or a subclass thereof.

Syntax:

isinstance(object, classinfo)

29. What relationship is appropriate for Course and Faculty?

→ The appropriate OOP (Object-Oriented Programming) relationship between Course and Faculty is an Association

One-to-Many: Typically, one Faculty member teaches multiple Courses, but each Course is associated with one Faculty member.

Many-to-Many: In more complex scenarios, both Course and Faculty can be associated with multiple instances of each other.

30. What relationship is appropriate for Student and Person?

→ In Object-Oriented Programming (OOP), the relationship between Student and Person is typically modeled as an inheritance relationship. This is because a Student is a specific type of Person, meaning that Student is a subclass of Person.

Module – 4 (Advance python programming)