**Q1. What is File function in python? What is keywords to create and write file.**

In Python, the open() function is commonly used to create, read, and write files. It is a built-in function that allows you to work with files in various modes, such as reading ('r'), writing ('w'), appending ('a'), and more. The open() function returns a file object that you can use to perform file operations.

***Opening File :***

file = open("filename.txt", "mode")

***FILE MODES :***

Read Mode (‘r’)

file = open("filename.txt", "r")

content = file.read()

**Q13. Explain Exception handling? What is an Error in Python?**

***EXCEPTON HANDLING:***

Exception handling is a mechanism in Python that deals with runtime errors. Errors in Python are exceptions that occur during the execution of a program. Exception handling allows you to gracefully handle these errors, preventing the program from terminating abruptly.

***Error in Python:***

An error in Python refers to a situation where the interpreter encounters a problem while executing the code. This can include syntax errors, logical errors, or runtime errors (exceptions).

**Q14. How many except statements can a try-except block have? Name Some built-in exception classes:**

***Number of Except Statements:***

A try-except block can have multiple except statements to handle different types of exceptions. Each except statement can handle a specific exception type.

***Built-in Exception Classes:***

ZeroDivisionError

ValueError

TypeError

FileNotFoundError

KeyError

IndexError

NameError

**Q15. When will the else part of try-except-else be executed?**

The else part of a try-except-else block is executed when no exceptions are raised in the try block. It provides a block of code to be executed if the try block executes successfully without any exceptions.

**Q16. Can one block of except statements handle multiple exceptions?**

Yes, a single block of except statements can handle multiple exceptions. You can either list multiple exception types in a single except statement or have multiple except statements for different exception types.

**Q17. When is the finally block executed?**

The finally block is always executed, regardless of whether an exception occurs or not. It is typically used for cleanup operations, such as closing files or releasing resources. The finally block is executed after the try block and any matching except blocks.

**Q18. What happens when '1' == 1 is executed?**

When '1' == 1 is executed, it checks for equality between a string "1" and an integer 1. Since the types are different, the result is False. In Python, the equality operator (==) checks for both value and type.

**Q19. How Do You Handle Exceptions With Try/Except/Finally In Python? Explain with coding snippets.**

```
try:
    # code that may raise exceptions
except SomeException as e:
    # handle the specific exception
finally:
    # code to be executed regardless of whether an exception occurred or not
```

**Q21. What are OOP concepts? Is multiple inheritance supported in Java?**

***OOP Concepts:***

Object-Oriented Programming (OOP) is a programming paradigm based on the concept of "objects," which can contain data in the form of fields (attributes or properties) and code in the form of procedures (methods). The four main OOP concepts are encapsulation, inheritance, polymorphism, and abstraction.

***Multiple Inheritance in Java:***

No, Java does not support multiple inheritance for classes. However, it supports multiple inheritance for interfaces, allowing a class to implement multiple interfaces.

**Q22. How to Define a Class in Python? What Is Self? Give an Example Of A Python Class.**

***Defining a Class:***

In Python, a class is defined using the class keyword. It is a blueprint for creating objects. Objects have member variables and have behavior associated with them in the form of methods.

***Self in Python:***

self is a reference to the instance of the class. It is the first parameter in every method, representing the instance on which the method is called.

**E.G.**

```
class Car:
    def __init__(self, make, model):
        self.make = make
        self.model = model
    def display_info(self):
        print(f"{self.make} {self.model}")
# Creating an instance of the class
my_car = Car("Toyota", "Camry")
# Accessing attributes and calling methods
print(my_car.make)
my_car.display_info()
```

In this example, Car is a class with attributes (make and model) and a method (display_info). The __init__ method is a special method called a constructor, and self refers to the instance of the class.

**Q25. Explain Inheritance in Python with an example? What is init? Or What Is A Constructor In Python?**

Inheritance allows a class (subclass/derived class) to inherit attributes and methods from another class (base class/parent class). It promotes code reusability and establishes an "is-a" relationship.

```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        pass  # Abstract method
class Dog(Animal):
    def speak(self):
        return f"{self.name} says Woof!"
```

```
class Cat(Animal):
    def speak(self):
        return f"{self.name} says Meow!"
# Example usage:
dog_instance = Dog("Buddy")
cat_instance = Cat("Whiskers")
print(dog_instance.speak())  # Output: Buddy says Woof!
print(cat_instance.speak())  # Output: Whiskers says Meow!
```

In Python, __init__ is a special method (constructor) in a class that gets called when an instance of the class is created. It initializes the attributes of the object. The self parameter refers to the instance of the class.

**Q26. What is Instantiation in terms of OOP terminology?**

Instantiation is the process of creating an instance (object) of a class. When you create an object of a class, you are said to instantiate the class.

**Q27. What is used to check whether an object o is an instance of class A?**

To check whether an object o is an instance of class A:

```
is_instance = isinstance(o, A)
```

**Q28. What relationship is appropriate for Course and Faculty?**

A relationship of association is appropriate. A faculty can be associated with multiple courses, and a course can be associated with multiple faculties.

**Q29. What relationship is appropriate for Student and Person?**

Inheritance is appropriate. A student is a type of person, so the Student class can inherit from the Person class.

**********
**********