Module – 4 (Advance python programming)

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

   The key function for working with files in Python is
   the open() function. The open() function takes two parameters; filename, and mode.

   There are four different methods (modes) for opening a file:
   "r" - Read - Default value. Opens a file for reading, error if the file does not exist
   "a" - Append - Opens a file for appending, creates the file if it does not exist
   "w" - Write - Opens a file for writing, creates the file if it does not exist
   "x" - Create - Creates the specified file, returns an error if the file exists

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

   →Explain Exception handling?

   Errors interrupt the flow of the program at the point where they appear, so any further code stops executing. This error is called an exception.

   Eg : n = int(input("Please enter a number: "))

   Please enter a number: 23.50
   Exception occurs like,
   ValueError: invalid literal for int() with base 10:'23.5'

→What is an Error in Python?

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

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

You can have multiple try blocks in a program but only one except statement with each try block.

Exception	**Description**
ArithmeticError	Raised when an error occurs in numeric calculations.
Exception	Base class for all exceptions.
KeyError	Raised when a key does not exist in a dictionary
NameError	Raised when a variable does not exist
SyntaxError	Raised when a syntax error occurs

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

The else block lets you execute code when there is no error.

→For example:

```
try:
  print("Hello")
except:
  print("Something went wrong")
else:
  print("Nothing went wrong")
```

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

Yes, a single block of except statements in Python can handle multiple exceptions. This feature allows you to handle different types of exceptions using a single block of code.

→For example:

Print one message if the try block raises a NameError and another for other errors:

```
try:
  print(x)
except NameError:
  print("Variable x is not defined")
except:
  print("Something else went wrong")
```

6. When is the finally block executed?

A "finally" clause is always executed regardless if an exception occurred.

→For example,

```
try:
    a=int(input("Enter value of A:"))
    b=int(input("Enter value of B:"))
```

```
        print("Sum:",a+b)
    except:
        print("Error!")
    finally:
        print("This is finally block!")
```

7. How Do You Handle Exceptions With Try/Except/Finally In Python?

First try clause is executed i.e. the code between try and except clause.

If there is no exception, then only try clause will run, except clause will not get executed.

If any exception occurs, the try clause will be skipped and except clause will run.

If any exception occurs, but the except clause within the code doesn’t handle it, it is passed on to the outer try statements.

If the exception is left unhandled, then the execution stops.

A try statement can have more than one except clause.

The finally block always executes after normal termination of try block or after try block terminates due to some exception.

Even if you return in the except block still the finally block will execute.

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

OOPS (Object-Oriented Programming System) is a programming paradigm that revolves around the concept of objects and classes. The main OOPS concepts are:

- **Encapsulation**: This concept binds together the data and the methods that manipulate that data, and keeps both safe from outside interference and misuse.
- **Abstraction**: This concept shows only the necessary information to the outside world while hiding the internal details.
- **Inheritance**: This concept allows one class to inherit the properties and behavior of another class.
- **Polymorphism**: This concept allows objects of different classes to be treated as objects of a common superclass.
- **Composition**: This concept allows objects to be composed of other objects or collections of objects.

Multiple Inheritance in Java:

Java does not support multiple inheritance in the classical sense. A Java class can only extend one parent class. However, Java does support multiple inheritance through interfaces. A Java class can implement multiple interfaces, which allows it to inherit behavior from multiple sources.

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

Defining a Class in Python:

In Python, a class is defined using the class keyword followed by the name of the class. The class definition consists of a block of code indented under the class header.

What is Self?

In Python, self is a reference to the current object being manipulated. It is an implicit argument that is passed to instance methods. When you call an instance method on an object, Python automatically passes the object as the first argument to the method, which is referred to as self.

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

Inheritance in Python:

Inheritance is a fundamental concept in object-oriented programming (OOP) that allows one class to inherit the properties and behavior of another class. In Python, inheritance is implemented using the class keyword followed by the name of the child class and the name of the parent class in parentheses.

What is __init__? Or What is a Constructor in Python?

In Python, __init__ is a special method that is called when an object is created from a class. It is used to initialize the attributes of the object. The __init__ method is also known as a constructor.
The __init__ method takes two arguments: self and any number of additional arguments. The self argument is a reference to the current object being manipulated. The additional arguments are used to initialize the attributes of the object.

11. What is Instantiation in terms of OOP terminology?

Class : A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members and methods, accessed via dot notation.
Class variable : A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods.

Data member : A class variable or instance variable that holds data associated with a class and its objects.

Function overloading : The assignment of more than one behavior to a particular function. The operation performed varies by the types of objects or arguments involved.

Instance variable : A variable that is defined inside a method and belongs only to the current instance of a class.

Inheritance : The transfer of the characteristics of a class to other classes that are derived from it.

Instance : An object created from a class. Each instance has its own set of attributes and methods.

Instantiation : The creation of an instance of a class.

Object : A unique instance of a data structure that's defined by its class.

Class Constructor: A special method within a class, often named __init__(), that is called during instantiation. It initializes the object's attributes and performs any necessary setup.