Experiment 1

Implementing Inheritance and Interface

Name: Sai Harsha Vardhan AVN

Roll-no: BCSE1823

Batch: ‘A’

Aim: Write a Java Program for the implementation of Inheritance and Interface

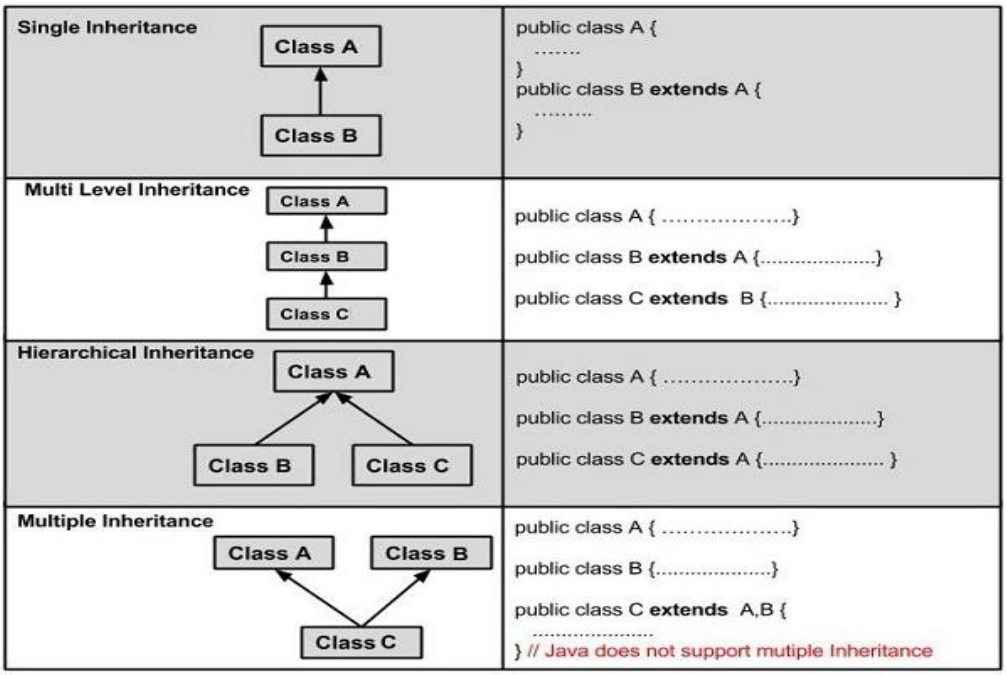
Theory:

Inheritance

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance, the information is made manageable in a hierarchical order.

The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).

Types of Inheritance is shown below:



Interface

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve abstraction*. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

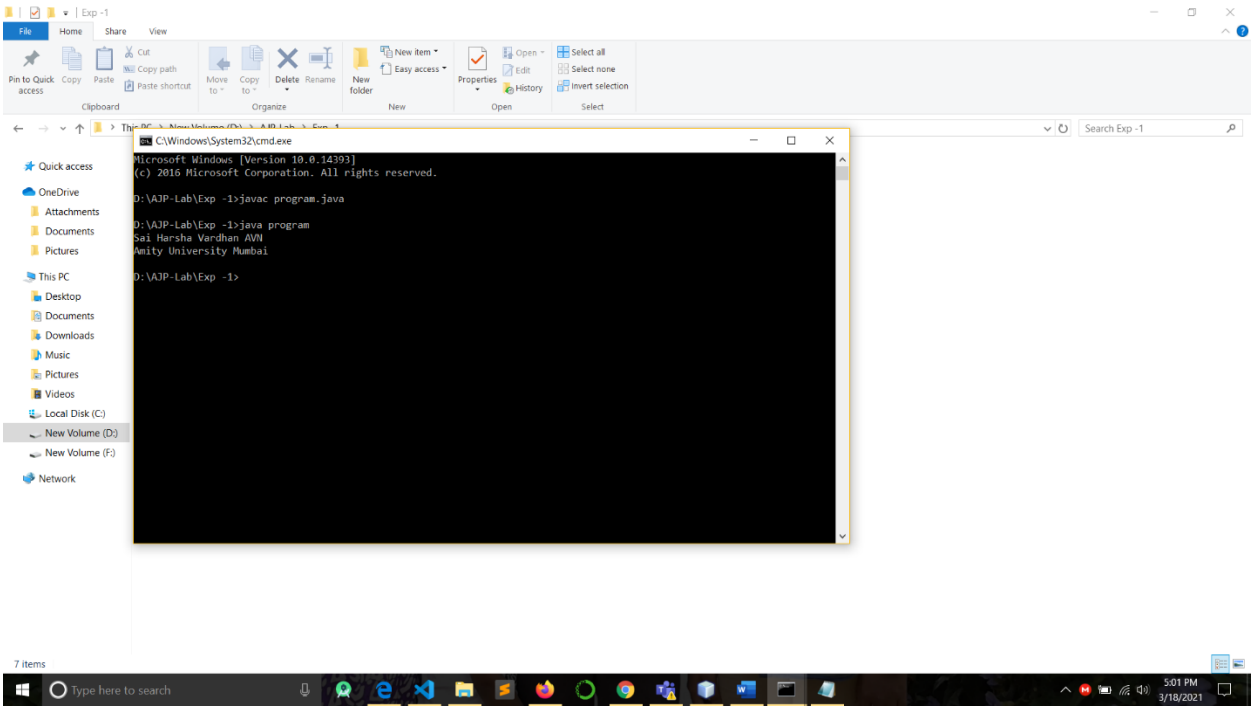
Code:

```
import java.io.*;
interface intfA
{
	void StudentName();
}
interface intfB extends intfA
{
	void StudentInstitute();
}
class program implements intfB
{
	public void StudentName()
	{
		System.out.println("Sai Harsha Vardhan AVN");
	}
	public void StudentInstitute()
	{
		System.out.println("Amity University Mumbai");
	}

	public static void main (String[] args)
	{
```

```
        program ob1 = new program();
        ob1.StudentName();
        ob1.StudentInstitute();
    }
}
```

Output:



Conclusion: Inheritance and Interface are implemented successfully.