2.7 Inheritance

This section will guide you to understand:

- What is Inheritance?
- Types of Inheritance in Java
- Why use Inheritance?
- Why is Multiple Inheritance not supported?

Development Environment:

- Eclipse
- Java 1.8

This guide has two subsections, namely:

2.7.1 Demonstrate types of inheritance

2.7.2 Push the code to your GitHub repositories

Step 2.7.1: Demonstrate types of inheritance

- There are various types of inheritance in Java:

  a) Single Inheritance:

  In Single Inheritance, one class extends another class (one class only).

```
Class A{
   public void methodA()
   {
     System.out.println("Base class method");
   }}
```

```
Class B extends A{
   public void methodB()
   {
     System.out.println("Child class method");
   }
   public static void main(String args[])
   {
     B obj = new B();
     obj.methodA(); //calling super class method
     obj.methodB(); //calling local method
  }
}
```

b) Multiple Inheritance:

In Multiple Inheritance, one class extends more than one class. Java does not support multiple inheritance.

c) Multi-level Inheritance:

In Multi-level Inheritance, one class can inherit properties from a derived class. Hence, the derived class becomes the base class for the new class.

```
Class X{
   public void methodX()
   {
     System.out.println("Class X method");
   }
}
Class Y extends X{
 public void methodY(){
 System.out.println("class Y method");
}
```

```
}
Class Z extends Y{
   public void methodZ()
   {
     System.out.println("class Z method");
   }
   public static void main(String args[])
   {
     Z obj = new Z();
     obj.methodX(); //calling grand parent class method
     obj.methodY(); //calling parent class method
     obj.methodZ(); //calling local method
  }
}
```

d) Hierarchical Inheritance:

In Hierarchical Inheritance, one class is inherited by many sub classes.

e) Hybrid Inheritance:

Hybrid inheritance is a combination of single and multiple inheritance.

Note: Java doesn't support hybrid/multiple inheritance.

Step 2.13.5: Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd java_program

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master