Name: Dhiraj Birajdar
Batch: 1154
Homework: Inheritance

Single Inheritance:

```
package inheritance;
class Animal {
   void eat() {
      System.out.println("Animal is eating.");
   }
}
class Dog extends Animal {
   void bark() {
      System.out.println("Dog is barking.");
   }
}
public class SingleInheritance {
   public static void main(String[] args) {
      Dog dog = new Dog();
      dog.eat();  // Inherited from Animal class
      dog.bark(); // Specific to Dog class
   }
}
```

Multilevel Inheritance:

```
package inheritance;
class A1 {
   void displayA1() {
      System.out.println("Class A1");
   }
}
class B1 extends A1 {
   void displayB1() {
```

```
        System.out.println("Class B1");
    }
}
class C1 extends B1 {
    void displayC1() {
        System.out.println("Class C1");
    }
}
public class MultilevelInheritance {
    public static void main(String[] args) {
        C1 obj = new C1();
        obj.displayA1();
        obj.displayB1();
        obj.displayC1();
    }
}
```

Multiple Inheritance:

```
package inheritance;
// Multiple inheritance (through interface)
interface A {
    void methodA();
}
interface B {
    void methodB();
}
class C implements A, B {
    public void methodA() {
        System.out.println("Implementing methodA");
    }
    public void methodB() {
        System.out.println("Implementing methodB");
```

```
    }
}
public class MultipleInheritance {
    public static void main(String[] args) {
        C c = new C();
        c.methodA();
        c.methodB();
    }
}
```

Hierarchical Inheritance:

```
package inheritance;
class Vehicle {
    void run() {
        System.out.println("Vehicle class");
    }
}
class Bike extends Vehicle {
    void ride() {
        System.out.println("Bike class");
    }
}
class Car extends Vehicle {
    void drive() {
        System.out.println("Car class");
    }
}
public class HierarchicalInheritance {
    public static void main(String[] args) {
        Bike Bike = new Bike();
        Bike.run();  // Inherited from Vehicle class
        Bike.ride(); // Specific to Bike class
```

```
        Car car = new Car();
        car.run();  // Inherited from Vehicle class
        car.drive(); // Specific to Car class
    }
}
```

Hybrid Inheritance:

```
package inheritance;
interface X {
    void methodA();
}
class Y {
    void methodB() {
        System.out.println("Class Y - methodB");
    }
}
class Z extends Y implements X {
    public void methodA() {
        System.out.println("Class Z - methodA");
    }
    void displayC() {
        System.out.println("Class Z - displayC");
    }
}
public class HybridInheritance {
    public static void main(String[] args) {
        Z c = new Z();
        c.methodA();
        c.methodB();
        c.displayC();
    }
}
```