Inheritance

Example 1

```
package Inheritance1;
/**
 *
 * @author Eshana
 */
public class Vehicle {
//Vihical attribute set to a protected access modifier
//If it was set to private, the Car class would not be able to access it.
protected String brand = "Ford"; // Vehicle attribute
public void honk() { // Vehicle method
System.out.println("Tuut, tuut!");
   }
}

package Inheritance1;
/**
 *
 * @author Eshana
 */
public class Car extends Vehicle {
private String modelName = "Mustang";    // Car attribute
   public static void main(String[] args) {

  // Create a myCar object
   Car myCar = new Car();

   // Call the honk() method (from the Vehicle class) on the myCar object
   myCar.honk();

   // Display the value of the brand attribute (from the Vehicle class) and the value of the modelName
from the Car class
   System.out.println(myCar.brand + " " + myCar.modelName);

   }

}
```

Example 2

```
package Example;

/**
 *
 * @author Eshana
 */
public class Student {

   String role ="Student";
   String faculty ="FOT";

   void does(){
      System.out.println("Study");
   }

}
```

```
package Example;

/**
 *
 * @author Eshana
 */
public class unistudent extends Student {

   String course ="IC 1009";

   public static void main(String[] args) {
    unistudent obj = new unistudent();
    System.out.println(obj.faculty);
    System.out.println(obj.course);
    System.out.println(obj.role);
    obj.does();
   }

}
```

Example 3

```
package Example;

/**
 *
 * @author Eshana
 */
public class Student {

   String role ="Student";
   String faculty ="FOT";

   void does(){
      System.out.println("Study");
   }

}
```

```
package Example;

/**
 *
 * @author Eshana
 */
public class unistudent extends Student {

   String course ="IC 1009";

   public static void main(String[] args) {
    unistudent obj = new unistudent();
    System.out.println(obj.faculty);
    System.out.println(obj.course);
    System.out.println(obj.role);
    obj.does();
   }

}
```

Example 4

//Single Inheritance

```
package SingleInheritance;
/**
 *
 * @author Eshana
 */
public class Animal {
   void eat(){
      System.out.println("eating");
   }
}
```

```
package SingleInheritance;
/**
 *
 * @author Eshana
 */

//Dog class inherits the Animal class
//So there is the single Inheritance
public class Dog extends Animal{

   void bark(){
      System.out.println("Barking..");
   }

}
```

```
package SingleInheritance;

/**
 *
 * @author Eshana
 */
public class TestInheritance {

   public static void main(String[] args) {
      Dog d = new Dog();
      d.bark();
      d.eat();
   }

}
```

Example 5

//Multilevel Inheritance

```
package multilevelInheritance;

/**
 *
 * @author Eshana
 */
public class Animal {
   void eat(){
      System.out.println("eating..");
   }

}
```

```
package multilevelInheritance;

/**
 *
 * @author Eshana
 */
public class BabyDog extends Dog{
   void weep(){
      System.out.println("Weeping..");
   }

}
```

```
package multilevelInheritance;

/**
 *
 * @author Eshana
 */
public class Dog extends Animal{
   void bark(){
      System.out.println("barking..");
   }

}
```

```
package multilevelInheritance;

/**
 *
 * @author Eshana
```

```
 */
public class TestInheritance2 {

   public static void main(String[] args) {
     BabyDog d=new BabyDog();
     d.weep();
     d.bark();
     d.eat();
   }

}
```

Example 6

```
package HierarchicalInheritance;

/**
 *
 * @author Eshana
 */
public class Animal {
   void eat(){
      System.out.println("eating..");
   }

}
```

```
package HierarchicalInheritance;

/**
 *
 * @author Eshana
 */
public class Cat extends Animal {
   void meow(){
      System.out.println("meow...");
   }

}
```

```
package HierarchicalInheritance;

/**
 *
 * @author Eshana
 */
```

```
public class Dog extends Animal {
   void bark(){
      System.out.println("barking..");
   }

}


package HierarchicalInheritance;

/**
 *
 * @author Eshana
 */
public class TestInheritance3 {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
    Cat c = new Cat();
    c.meow();
    c.eat();
   }

}
```