Assignment 2

Deadline: 15/7/2019

Task 1:

Write down the outputs for Tester Class written below:

```
class Alfie extends John {
  String power = "Jewsish leader";
  public void method1() {
    super.method1();
    System.out.println("Gunman m1");
    System.out.println(this);
  }
  public void method3() {
    System.out.println("Gunman m2");
  }
}

class Arthur extends TommyShelby {
  String power = "Arthur Shelby";
  public void method1() {
    System.out.println("mercenary m1");
  }
  public void method2() {
    System.out.println("mercenary m2");
  }
  public void method3() {
    System.out.println("mercenary m3");
    method1();
  }
}
```

```
class Polly extends TommyShelby {
  String power = "Polly Gray ";
  public void method2() {
    System.out.println("Polly m2");
  }
  public void method3() {
    System.out.println("Polly m3");
  }
  public String toString(){
    return "The Accountant is "+ power;
  }
}

class TommyShelby{
  String power = "Thomas Shelby";
  public void method1() {
    method2();
    System.out.println("Godfather m1");
  }
  public void method2() {
    System.out.println("Godfather m2");
  }
  public String toString(){
    method2();
    return "Godfather is"+ power;
  }
  public String toString(){
    return "The elder brother is "+ power;
  }
}

class John extends Arthur {
  String power = "hotShot";
  public void method1() {
    System.out.println("Jonny m1");
```

```
  }
  public void method4() {
    System.out.println("Jonny m4");
  }
  public String toString(){
    method2();
    return "Johny gets married because he is "+ power;
  }
}

public class Tester1{
  public static void main(String[] args){
    TommyShelby  don1   = new TommyShelby();
    TommyShelby  don2   = new Arthur();
    Polly newYorker1 = new Polly();
    Arthur hotHeaded1 = new Arthur();
    Arthur hotHeaded2 = new John();
    Object obj1 = new Polly();
    TommyShelby newSicilian = new Alfie();

    ((Arthur)hotHeaded2).method3();
    System.out.println(newYorker1);
    ((TommyShelby)newYorker1).method2();
  ((Object)don1).toString();
   don2.method2();
  System.out.println(((Arthur)hotHeaded2).power);
    ((TommyShelby)newYorker1).method3();
   hotHeaded1.method1();
   ((John)obj1).method2();
   System.out.println(newSicilian);
  }
}
```

Task 2:

```
class Frodo extends Bilbo{
    public String name = "Frodo";
    public void method1(){
        System.out.println(super.name+" 1");
        System.out.println("Frodo 1");
        super.method1();
    }
    public void method3(){
        System.out.println(this.name+" 3");
        System.out.println("Frodo 3");
    }
}
class Gandalf{
    public String name = "Gandalf";
    public void method1(){
        System.out.println("Gandalf 1");
    }
    public void method2(){
        System.out.println("Gandalf 2");
        method1();
        System.out.println(name+" 2");
    }
}
class Bilbo extends Gandalf{
    public void method1(){
        System.out.println("Bilbo 1");
        System.out.println(super.name+" 1");
    }
}
class Gollum extends Gandalf{
    public String name = "Gollum";
    public void method3(){
        System.out.println("Gollum 3");
        this.method2();
    }
}
```

And assuming the following variables have been defined:

Gandalf var1 = new Frodo(); Gandalf var2 = new Bilbo(); Gandalf var3 = new Gandalf(); Object var4 = new Bilbo(); Bilbo var5 = new Frodo(); Object var6 = new Gollum();

In the table below, indicate in the right-hand column the output produced by the statement in the left-hand column. If the statement produces more than one line of output, indicate the line breaks with slashes as in "a/b/c" to indicate three lines of output with "a" followed by "b" followed by "c". If the statement causes an error, fill in the right-hand column with either the phrase "compiler error" or "runtime error" to indicate when the error would be detected.

Statement	Output
`System.out.println(var1.name);`	

`System.out.println(var2.name);`	
`System.out.println(var3.name);`	

`System.out.println(var4.name);`	
`System.out.println(var5.name);`	
`System.out.println(var6.name);`	
`var1.method1();`	
`var2.method1();`	
`var4.method1();`	
`var6.method1();`	
`var1.method2();`	
`var3.method2();`	
`var4.method2();`	
`var5.method2();`	
`var6.method2();`	
`((Frodo)var4).method3();`	
`((Frodo)var6).method2();`	
`((Gollum)var1).method3();`	
`((Gollum)var4).method1();`	
`((Gandalf)var1).method2();`	
`((Frodo)var4).method1();`	
`((Gollum)var6).method2();`	
`((Gandalf)var2).method1();`	
`((Bilbo)var6).method2();`	
`((Frodo)var1).method3();`	

`((Gandalf)var5).method3();`	