

```
1 package Questions;
2
3 final class NewClass{
4     int a;
5     int b;
6 }
7 public class FinalClass extends NewClass{
8     public static void main(String[] args) {
9         System.out.println("Hello World");
10    }
11 }
12
```

```
1 package Questions;
2
3 class FinalShape{
4     public final void print(){
5         System.out.println("This is a final shape"
6     );
7 }
8 public class FinalMethods extends FinalShape{
9     public void print(){
10         System.out.println("This is inherited shape
11 ");
12     }
13     public static void main(String[] args) {
14         FinalMethods fm = new FinalMethods();
15         fm.print();
16     }
17 }
18
```

```
1 package Questions;
2
3 abstract class AbstractShape{ // Private member
4     variable private
5     String color;
6     // Constructor
7     public AbstractShape (String color) {
8         this.color = color;
9     }
10    public String toString() {
11        return "Shape of color=\"" + color + "\"";
12    }
13    // All Shape subclasses must implement a method
14    called getArea()
15    abstract public double getArea();
16 }
17 public class AbstractClass {
18     public static void main(String[] args) {
19         AbstractShape s1 = new Rectangle("red"
20         , 4, 5);
21         System.out.println(s1);
22         System.out.println("Area is " + s1.
23         getArea());
24         AbstractShape s2 = new Triangle("blue"
25         , 4, 5);
26         System.out.println(s2);
27         System.out.println("Area is " + s2.
28         getArea());
29         // Cannot create instance of an abstract class
30         AbstractShape s3 = new AbstractShape("
31         green");// Compilation Error!!
32     }
33 }
```

```

1 package Questions;
2
3 class Shape {
4     private String color;
5     public Shape (String color) {
6         this.color = color;
7     }
8     public String toString() {
9         return "Shape of color=\"" + color + "\""
10    ; }
11    public double getArea() {
12        System.err.println("Shape unknown! Cannot
13        compute area!"); return 0; // Need a return to
14        compile the program
15    } }
16
17 class Rectangle extends Shape { // Private member
18    variables
19    private int length;
20    private int width;
21    public Rectangle(String color, int length, int
22    width) {
23        super(color);
24        this.length = length;
25        this.width = width;
26    }
27    public String toString() {
28        return "Rectangle of length=" + length + "
29        and width=" + width + ", subclass of " + super.
30        toString(); }
31    public double getArea() { return length*
32    width; }
33 }
34
35 class Triangle extends Shape { // Private member
36    variables private int base;
37    private int height;
38    private int base;
39    // Constructor
40    public Triangle(String color, int base, int
41    height) {

```

```
32         super(color);
33         this.base = base;
34         this.height = height;
35     }
36     public String toString() {
37         return "Triangle of base=" + base + " and
height=" + height + ", subclass of " + super.
toString();
38     }
39     public double getArea() { return 0.5*base*
height; }
40 }
41
42
43 public class MethodOverriding {
44     public static void main(String[] args) {
45         Shape s1 = new Rectangle("red", 4, 5);
46         System.out.println(s1);
47         System.out.println("Area is " + s1.getArea
());
48         Shape s2 = new Triangle("blue", 4, 5);
49         System.out.println(s2);
50         System.out.println("Area is " + s2.getArea
());
51     }
52 }
53
```

```
1 package Questions;
2 //exercise 1.2
3 public class SingleInheritanceMain {
4     public static void main(String[] args) {
5         E e = new E();
6         e.show();
7     }
8 }
9
10 class A{
11     public int a =100;
12 }
13
14 class B extends A{
15     public int a = 80;
16 }
17 class C extends B{
18     public int a = 60;
19 }
20
21 class D extends C{
22     public int a = 40;
23 }
24
25 class E extends D{
26     public int a = 10;
27     public void show(){
28         System.out.println("E: "+this.a);
29         System.out.println("D: "+((D) this).a);
30         System.out.println("C: "+((C) this).a);
31         System.out.println("B: "+((B) this).a);
32         System.out.println("A: "+((A) this).a);
33     }
34 }
35
```

```
1 package Questions;
2
3
4 class Room {
5     double length,breadth,height;
6     public Room(){
7         length=-1;
8         breadth=-1;
9         height=-1;
10    }
11
12    public Room(double l,double b,double h) {
13        length=l;
14        breadth=b;
15        height=h;
16    }
17    Room(double len) { // Single parametrised
18        constructor
19        length=breadth=height=len;
20    }
21    double volume() {
22        return length*breadth*height;
23    }
24 }
25 public class ConstructorOverloading {
26     public static void main(String args[]) {
27         Room a = new Room(20, 30, 40);
28         Room b = new Room();
29         Room c = new Room(10);
30         double vol;
31         vol = a.volume();
32         System.out.println("Volume of room a is "
33         + vol);
34         vol = b.volume();
35         System.out.println("Volume of room b is "
36         + vol);
37         vol = c.volume();
38         System.out.println("Volume of room c is "
39         + vol);
40     }
41 }
```

```
38 }
```

```
39
```