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

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

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

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

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

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

```
1 package Questions;
 2
 3
 4 class Room {
 5
       double length, breadth, height;
       public Room(){
 6
 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
   constructor
18
            length=breadth=height=len;
19
       }
20
21
       double volume() {
22
           return length*breadth*height;
23
       }
24 }
25 public class ConstructorOverloading {
       public static void main(String args[]) {
26
           Room a = new Room(20, 30, 40);
27
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 "
    + vol);
33
           vol = b.volume();
           System.out.println("Volume of room b is "
34
    + vol);
35
           vol = c.volume();
36
           System.out.println("Volume of room c is "
    + vol);
37
       }
```