```java
package Exercises;

interface Movable{
    void moveUp();
    void moveDomn();
    void moveRight();
    void moveLeft();
}

class MovablePoint implements Movable{

    int x;
    int y;
    int xSpeed;
    int ySpeed;

    public MovablePoint(int x, int y, int xSpeed, int ySpeed) {
        this.x = x;
        this.y = y;
        this.xSpeed = xSpeed;
        this.ySpeed = ySpeed;
    }

    @Override
    public String toString() {
        return "MovablePoint{" +
                "x=" + x +
```

```
28                      ", y=" + y +
29                      ", xSpeed=" + xSpeed
    +
30                      ", ySpeed=" + ySpeed
    +
31                      '}';
32      }
33
34      @Override
35      public void moveUp() {
36          y-=ySpeed;
37      }
38
39      @Override
40      public void moveDomn() {
41          y+=ySpeed;
42      }
43
44      @Override
45      public void moveRight() {
46          x+=xSpeed;
47      }
48
49      @Override
50      public void moveLeft() {
51          x-=xSpeed;
52      }
53
54      public void compare(MovablePoint
```

```
54  test){
55          if (this.x == test.x && this
    .y == test.y){
56              System.out.println("Both
     are at same position");
57          }
58
59          //          To compare the X
    coordinate
60          if (this.x > test.x){
61              System.out.println("
    First one is right side of second");
62          } else if (this.x < test.x){
63              System.out.println("
    First one is left sied of second");
64          } else{
65              System.out.println("Both
     have same x coordinate");
66          }
67
68  //          To compare the Y
    coordinate
69          if (this.y > test.y){
70              System.out.println("
    First one is above second");
71          } else if (this.y < test.y){
72              System.out.println("
    First one is below second");
73          } else{
```

```
74                System.out.println("
   Both have the same Y coordinate");
75            }
76        }
77    }
78
79    class MovableCircle implements
   Movable{
80
81        private MovablePoint center;
82        private int radius;
83
84        public MovableCircle(int x, int
    y, int xSpeed, int ySpeed, int
   radius) {
85            super();
86            center = new MovablePoint(x
   ,y,xSpeed,ySpeed);
87            this.radius = radius;
88        }
89
90        @Override
91        public String toString() {
92            return "MovableCircle{" +
93                    "center=" + center.
   toString() +
94                    ", radius=" +
   radius +
95                    '}';
```

```
 96      }
 97
 98      public void moveUp() {
 99          center.y-=center.ySpeed;
100      }
101
102      @Override
103      public void moveDomn() {
104          center.y+=center.ySpeed;
105      }
106
107      @Override
108      public void moveRight() {
109          center.x+=center.xSpeed;
110      }
111
112      @Override
113      public void moveLeft() {
114          center.x-=center.xSpeed;
115      }
116
117      void compareRadius(
     MovableCircle temp){
118          if (this.radius > temp.
     radius){
119              System.out.println("
     First circle has bigger radius");
120          } else if(this.radius <
     temp.radius){
```

```
121                 System.out.println("
    Second circle has bigger radius");
122             } else{
123                 System.out.println("
    Both circles have same radius");
124             }
125         }
126 
127         void compareCenter(
    MovableCircle temp){
128             this.center.compare(temp.
    center);
129         }
130 
131         void compare(MovableCircle temp
    ){
132             this.compareRadius(temp);
133             this.center.compare(temp.
    center);
134         }
135 }
136 public class Exercise1 {
137         public static void main(String
    [] args) {
138             Movable m1 = new
    MovablePoint(5, 6, 10, 10);      //
    upcast
139              System.out.println(m1);
140              m1.moveLeft();
```

```
141             System.out.println(m1);
142             Movable m2 = new
    MovableCircle(2, 1, 2, 20, 10); //
    upcast
143             System.out.println(m2);
144             m2.moveRight();
145             System.out.println(m2);
146         }
147 }
148
```

```java
package Exercises;

import java.util.Arrays;

interface BinaryInterface{
    void toBinary(int N);

}

class OneComplement implements BinaryInterface{
    int[] Arr = new int[8];

    public void toBinaryComplement(int N) {
        if (N > -128 && N < 128){
            String bin;
            bin = Integer.toBinaryString(N);
            int binLength = bin.length();
            System.out.println(binLength);
            if (N < 0){
                for (int i = 0; i < 8; i++) {
                    Arr[Arr.length-i-1] = Integer.parseInt(String.valueOf(bin.charAt(binLength-i-1)));
```

```
22                  }
23              } else if (N > 0){
24                  for (int i = 0; i <
    binLength; i++) {
25                      Arr[Arr.length-i
    -1] = Integer.parseInt(String.
    valueOf(bin.charAt(binLength-i-1)));
26                  }
27              }
28
29              for (int i = 0; i < Arr.
    length; i++) {
30                  if (Arr[i] == 0){
31                      Arr[i] = 1;
32                  } else{
33                      Arr[i] = 0;
34                  }
35              }
36              System.out.println(
    Arrays.toString(Arr));
37          } else{
38              System.out.println("Not
    an 8 bit number");
39          }
40      }
41
42      @Override
43      public void toBinary(int N) {
44          if (N > -128 && N < 128){
```

```
45              String bin;
46              bin = Integer.
   toBinaryString(N);
47              int binLength = bin.
   length();
48              System.out.println(
   binLength);
49              if (N < 0){
50                  for (int i = 0; i <
   8; i++) {
51                      Arr[Arr.length-i
   -1] = Integer.parseInt(String.
   valueOf(bin.charAt(binLength-i-1)));
52                  }
53              } else if (N > 0){
54                  for (int i = 0; i <
   binLength; i++) {
55                      Arr[Arr.length-i
   -1] = Integer.parseInt(String.
   valueOf(bin.charAt(binLength-i-1)));
56                  }
57              }
58              System.out.println(
   Arrays.toString(Arr));
59          } else{
60              System.out.println("Not
   an 8 bit number");
61          }
62      }
```

```
63
64
65 }
66
67
68
69 public class Exericse2 {
70     public static void main(String
   [] args) {
71         OneComplement o = new
   OneComplement();
72         o.toBinary(127);
73 //         System.out.println(Integer
   .toBinaryString());
74         System.out.println();
75     }
76 }
77
```