```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 +
                "', y=" + y +
                ", xSpeed=" + xSpeed +
                ", ySpeed=" + ySpeed +
                '}';
    }

    @Override
    public void moveUp() {
        y-=ySpeed;
    }

    @Override
    public void moveDomn() {
        y+=ySpeed;
    }

    @Override
    public void moveRight() {
        x+=xSpeed;
    }

    @Override
    public void moveLeft() {
        x-=xSpeed;
    }

    public void compare(MovablePoint
 test){
        if (this.x == test.x && this.y == test.y){
            System.out.println("Both are at same position");
        }

        //        To compare the X coordinate
        if (this.x > test.x){
            System.out.println("First one is right side of second");
        } else if (this.x < test.x){
            System.out.println("First one is left sied of second");
        } else{
            System.out.println("Both have same x coordinate");
        }

//        To compare the Y coordinate
        if (this.y > test.y){
            System.out.println("First one is above second");
        } else if (this.y < test.y){
            System.out.println("First one is below second");
        } else{
            System.out.println("Both have the same Y coordinate");
        }
    }
}

class MovableCircle implements Movable{

    private MovablePoint center;
    private int radius;

    public MovableCircle(int x, int y, int xSpeed, int ySpeed, int radius) {
        super();
        center = new MovablePoint(x,y,xSpeed,ySpeed);
        this.radius = radius;
    }

    @Override
    public String toString() {
        return "MovableCircle{" +
                "center=" + center.toString() +
                ", radius=" + radius +
                '}';
```

```java
    }

    public void moveUp() {
        center.y-=center.ySpeed;
    }

    @Override
    public void moveDomn() {
        center.y+=center.ySpeed;
    }

    @Override
    public void moveRight() {
        center.x+=center.xSpeed;
    }

    @Override
    public void moveLeft() {
        center.x-=center.xSpeed;
    }

    void compareRadius(MovableCircle temp){
        if (this.radius > temp.radius){
            System.out.println("First circle has bigger radius");
        } else if(this.radius < temp.radius){
            System.out.println("Second circle has bigger radius");
        } else{
            System.out.println("Both circles have same radius");
        }
    }

    void compareCenter(MovableCircle temp){
        this.center.compare(temp.center);
    }

    void compare(MovableCircle temp){
        this.compareRadius(temp);
        this.center.compare(temp.center);
    }
}
public class Exercise1 {
    public static void main(String[] args) {
        Movable m1 = new MovablePoint(5, 6, 10, 10);     // upcast
         System.out.println(m1);
         m1.moveLeft();
        System.out.println(m1);
        Movable m2 = new MovableCircle(2, 1, 2, 20, 10); // upcast
        System.out.println(m2);
        m2.moveRight();
        System.out.println(m2);
    }
}

```

```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)));
                }
            } else if (N > 0){
                for (int i = 0; i < binLength; i++) {
                    Arr[Arr.length-i-1] = Integer.parseInt(String.valueOf(bin.charAt(binLength-i-1)));
                }
            }

            for (int i = 0; i < Arr.length; i++) {
                if (Arr[i] == 0){
                    Arr[i] = 1;
                } else{
                    Arr[i] = 0;
                }
            }
            System.out.println(Arrays.toString(Arr));
        } else{
            System.out.println("Not an 8 bit number");
        }
    }

    @Override
    public void toBinary(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)));
                }
            } else if (N > 0){
                for (int i = 0; i < binLength; i++) {
                    Arr[Arr.length-i-1] = Integer.parseInt(String.valueOf(bin.charAt(binLength-i-1)));
                }
            }
            System.out.println(Arrays.toString(Arr));
        } else{
            System.out.println("Not an 8 bit number");
        }
    }
```

```java


}



public class Exericse2 {
    public static void main(String[] args) {
        OneComplement o = new OneComplement();
        o.toBinary(127);
//        System.out.println(Integer.toBinaryString());
        System.out.println();
    }
}

```