Activity 4-1

CAD

1. Design a class named **TwoDPoint** to represent a point with x and y coordinates.
2. The class contains:
   - Two private data fields **x** and **y** that represent the coordinates
   - A no-arg constructor that constructs a point (0,0)
   - A constructor that constructs a point with specific coordination
   - Two **get** methods for data x and y, respectively
   - A method named **getDistance** that returns the distance from current point to another point of the **TwoDPoint** type. The method signature is:

     ```
     public double getDistance(TwoDPoint remotePoint)
     ```
3. Design another class named **ThreeDPoint t**o represent a point with x, y and z coordinates.
4. The class:
   - Is a subclass of **TwoDPoint**.
   - Has a private data fields **z** to represent the coordinates
   - Has a no-arg constructor that constructs a 3D point (0,0,0)
   - Has a constructor that constructs a point with specific (x,y,z) coordination
   - A **get** method for data z.
   - A method named **getDistance** that returns the distance from current point to another point of the **ThreeDPoint** type. Try to reuse the logic in **getDistance() i**n the parent class.

Activity 4-2

Flower

1. The test program creates a Rose object to represents a Pink rose name Majolica, originally from Damascus and it has a meaning of Happiness.
2. Complete the constructor for the class Rose.

```
public static void main(String[] args) {
   Rose r = new Rose ("Pink","Damascus","Majolica","Happiness");
}
```

```
class Flower {
    String color;
    String origin;

    public Flower(String color, String origin) {
        this.color = color;
        this.origin = origin;
    }
}
```

```
class Rose extends Flower {
   String name;
   String meaning;

   // complete the constructor

}
```

Activity 4-3

Travel

1. The test program creates a Boat object. Complete the class Boat to produce the following output for the test program:

```
Moving in Sea at speed of 50
The base port for WhiteSand is Raffles Marina
```

2. Indicate which statement in the test program exhibits the polymorphism.

```
public class Test {
    public static void main(String[] args) {
        Boat b  = new Boat("50","WhiteSand", "Sea", "Raffles Marina");
        setModeOfTravel(b);
    }
    public static void setModeOfTravel(Transportation t){
        t.move();
    }
}
```

```
class Transportation {
    String speed;
    String name;
    String type; // land, sea or air

    public Transportation(String speed, String name, String type) {
        this.speed = speed;
        this.name = name;
        this.type = type;
    }

    void move(){
        System.out.println ("Moving in "+ type + " at speed of " + speed);
    }
}
```

```
 class Boat extends Transportation {
    String basePort;

     // complete the class
 }
```

Activity 4-4

Boat Farm

1. Continue with class Boat from previous question.
2. Create a few Boat objects.
3. Store them in an ArrayList.
4. Write a test program with comparator to sort the Boat objects by their name.
5. Code suitable display to test the program.

Activity 4-5

Maximum

1. Complete the generic method getMaximum which takes in an ArrayList of elements, x.
2. The method returns the maximum of the elements.
3. Assuming that you only need to take care of elements of String, Long, Integer, Double, Byte, Short, Character and Double.
4. The operator > and < cannot apply to generic. Need to use **compareTo()** for comparison.

```
public class Test {
  public static void main(String[] args) {
    ArrayList<Character> a1 = new ArrayList( Arrays.asList('r','P','t','w'));
    System.out.println (getMaximum(a1));
    ArrayList<Float> a2 = new ArrayList( Arrays.asList(111f,-99f,888f,333f));
    System.out.println (getMaximum(a2));
    ArrayList<String> a3 = new ArrayList( Arrays.asList("SP","SG","EEE","DCPE"));
    System.out.println (getMaximum(a3));
  }

  public static <E extends Comparable<E>> ______ getMaximum(_________ x)
  {
      // complete the code to return the maximum
  }
}
```