CS2030S Programming Methodology II
Semester 2 2021/2022

26 & 27 January 2022
Problem Set #1

1. Consider the following definition of a `Vector2D` class:

```
class Vector2D {
    private double x;
    private double y;

    Vector2D(double x, double y) {
        this.x = x;
        this.y = y;
    }

    void add(Vector2D v) {
        this.x = this.x + v.x;
        this.y = this.y + v.y;
        // line A
    }
}
```

   (a) Suppose that the following program fragment is in a `main` method, show the content of the stack and the heap when the execution reaches the line labelled `A` above.

```
Vector2D v1 = new Vector2D(1, 1);
Vector2D v2 = new Vector2D(2, 2);
v1.add(v2);
```

   Label your variables and the values they hold clearly. You can use arrows to indicate object references. Draw boxes around the stack frames of the methods `main` and `add`, and label them.

   (b) Suppose that the representation of `x` and `y` have been changed to a double array:

```
class Vector2D {
    private double[] coord2D;

    ...
}
```

      i. What changes do you need for the other parts of class `Vector2D`
      ii. Would the program fragment in 1a above be valid?
      Show the content of the stack and the heap when the execution reaches the line labelled `A` again.

2. Study the following `Point` and `Circle` classes.

```
public class Point {
    private double x;
    private double y;

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }
}

public class Circle {

    private Point centre;
    private int radius;

    public Circle(Point centre, int radius) {
        this.centre = centre;
        this.radius = radius;
    }

    @Override
    public boolean equals(Object obj) {
        System.out.println("equals(Object) called");
        if (obj == this) {
            return true;
        }
        if (obj instanceof Circle) {
            Circle circle = (Circle) obj;
            return (circle.centre.equals(centre) && circle.radius == radius);
        } else {
            return false;
        }
    }

    public boolean equals(Circle circle) {
        System.out.println("equals(Circle) called");
        return circle.centre.equals(centre) && circle.radius == radius;
    }
}
```

Given the following program fragment,

```
Circle c1 = new Circle(new Point(0, 0), 10);
Circle c2 = new Circle(new Point(0, 0), 10);
Object o1 = c1;
Object o2 = c2;
```

what is the output of the following statements?

(a) `o1.equals(o2);`

(b) `o1.equals((Circle) o2);`

(c) `o1.equals(c2);`

(d) `o1.equals(c1);`

(e) `c1.equals(o2);`

(f) `c1.equals((Circle) o2);`

(g) `c1.equals(c2);`

(h) `c1.equals(o1);`