Generics

1. Read the code and answer the questions on the next page.

```
import java.util.ArrayList;
import java.lang.Number;

abstract class GeometricObject {
	private String name;

	/** Construct a default geometric object */
	protected GeometricObject(String name) {
		this.name = name;
	}

	@Override
	public String toString() {
		return name;
	}

	/** Abstract method getArea */
	public abstract double getArea();
}

class Circle extends GeometricObject implements Comparable<Circle>{
	private double radius;

	public Circle (String name, double radius) {
		super(name);
		this.radius = radius;
	}

	/** Return area */
	public double getArea() {
		return radius * radius * Math.PI;
	}

	public int compareTo(Circle other) {
		if(this.getArea()> other.getArea())
			return 1;
		else if (this.getArea()==other.getArea())
			return 0;
		else
			return -1;
	}
}

public class GeometryTester {
	public static void main(String[] args) {
		Circle c1 = new Circle("arctic", 10);
		Circle c2 = new Circle("equator", 30);
		Circle c3 = new Circle("capricorn", 20);

		ArrayList<Circle> objects = new ArrayList<>();
		objects.add(c1);
		objects.add(c2);
		objects.add(c3);
```

```
                System.out.println("and the Emmy goes to: " + max1(objects));

        Does the above statement compile? _________________ (why/why not?)

        What does it print if it compiles?_________________

                System.out.println("and the Oscar goes to: " + max2(objects));

        Does the above statement compile? _________________ (why/why not?)

        What does it print if it compiles?_________________
        }

        public static GeometricObject max1(ArrayList<GeometricObject> objects) {
                if(objects.isEmpty())
                        return null;
                GeometricObject current = objects.get(0);
                for(int i=1; i<objects.size(); i++) {
                        GeometricObject temp = objects.get(i);
                        if(current.getArea() < temp.getArea())
                                current = temp;
                }
                return current;
        }

        public static <E extends GeometricObject> E max2(ArrayList<E> objects) {
                if(objects.isEmpty())
                        return null;
                E current = objects.get(0);
                for(int i=1; i<objects.size(); i++) {
                        E temp = objects.get(i);
                        if(current.getArea() < temp.getArea())
                                current = temp;
                }
                return current;
        }
}
```

2. What is printed?

```
public class TwoGenerics <T, K>{
        T al;
        K sl;
        public TwoGenerics (T al, K sl) {
                this.al = al;
                this.sl = sl;
        }
        public void printer (T a, K b){
                System.out.println(a + " " + b);
        }
        public static void main (String[] args) {
                TwoGenerics<Integer, Double> t = new TwoGenerics<>(new Integer(4), new Double(5.0));
                TwoGenerics<Character, Boolean> t2 = new TwoGenerics<>(new Character ('a'), new Boolean (true));

                t.printer(t.al, t.sl);
                t2.printer(t2.al, t2.sl);
        }
}
```