Advanced Placement Computer Science **Unit 9 Review**

Part I: Multiple Choice (12 questions / 3 points each)

1) Given a written description of a programming project, you must choose the best design. (super class & subclasses)

2) Know how many classes a given class can extend.

3) Understand the following terms:
   - an inherited method
   - an overridden method
   - an overloaded method
   - an abstract method

4) Given an inheritance hierarchy, select a false statement about the classes.

5) Given a super class and a subclass: know how to complete the parametric constructor of the subclass.

6)
```
public class Circle
{
   private int radius;

   public Circle(int r)
   { ... }
   public int getRadius()
   { ... }
}

public class Cylinder extends Circle
{
   private int height;

   public Cylinder(int r, int h)
   { ... }
   public double area()
   { ... }
}
```

Which of the following code segments will not cause an error?

- ```
  Cylinder c1 = new Circle(3);
  int myRad = c1.getRadius();
  ```

- ```
  Circle c2 = new Cylinder(5,8);
  int myRad = c2.getRadius();
  ```

- ```
  Circle c3 = new Cylinder(6,8);
  double myArea = c3.area();
  ```
  ```

7)
```
public class SuperClass
{
   public int A()
   {
         return 2;
   }
   public int B()
   {
         return 3;
   }
}

public class SubClass extends SuperClass
{
   public int A()
   {
         return super.B() + 5;
   }
   public int B()
   {
         return super.A() + 10;
   }
}
```

Assume the following code appears in a Driver class, what would be printed out?

```
SuperClass x = new SubClass();
System.out.println(x.B());
```

8)
```
public abstract class Quad
{ ... }

public class Trapezoid extends Quad
{ ... }
```

Which of the following lines of code will not cause an error?

- `Quad q = new Trapezoid();`
- `Quad q = new Quad();`
- `Trapezoid t = new Quad();`

9)
```
public class Degree
{ ... }

public class Phd extends Degree
{ ... }
```

Assume the following code segment appears in a Driver class:

```
ArrayList<Degree> list = new ArrayList<Degree>();
Phd p = new Phd();
Degree d = new Degree();
list.add(p);
list.add(d):
```

Which of the following lines will cause an error?

```
Degree x = list.get(0);
Degree x = list.get(1);
Phd y = list.get(0);
Phd y = list.get(1);
```

10) Given an abstract class and a subclass that extends it: be able to select a false statement about the abstract class.

- Must an abstract class be extended to be used?
- Does a subclass have direct access to the private instance fields of its super class?

11) Given an abstract class and a subclass that extends it: be able to select a true statement about the subclass.

- Can a second subclass use methods of the first subclass?
- Does the abstract method of the super class have to have the exact same name in the subclass?

12) Given an abstract class, 2 subclasses that extend it, and a driver class that fills an ArrayList with objects of both subclasses: How would you call the abstract method of each object as it is removed from the ArrayList? (with or without the use of an enhanced for loop)

Part II: Free Response

```
public class Point
{
      private int xCoord;
      private int yCoord;
      public Point(int x, int y)
      {
            xCoord = x;
            yCoord = y;
      }
      public int getX()
      {
            return xCoord;
      }
      public int getY()
      {
            return yCoord;
      }
}

public abstract class Quadrilateral
{
      private String label;   // i.e. "ABCD"
      public Quadrilateral(String lbl)
      {
            label = lbl;
      }
      public String getLabel()
      {
            return label;
      }
      protected double distance(Point one, Point two)
      {
            int x1 = one.getX();
            int y1 = one.getY();
            int x2 = two.getX();
            int y2 = two.getY();
            double ans = Math.sqrt(Math.pow(x2-x1,2) + Math.pow(y2-y1,2));
            return ans
      }
      public abstract double perimeter();
      public abstract double area();
}
```

```
public class Parallelogram extends Quadrilateral
{
	private Point topLeft;
	private Point botLeft;
	private Point botRight;
	public Parallelogram(String lbl, Point tL, Point bL, Point bR)
	{
		<for you to complete 4 points>
	}
	public double perimeter()
	{
		<for you to complete 8 points>
	}
	public double area()
	{
		<for you to complete 8 points>
	}
}
public class Trapezoid extends Quadrilateral
{
	private Point topLeft;
	private Point botLeft;
	private Point botRight;
	private Point topRight;

	public Trapezoid(String lbl, Point tL, Point bL, Point bR, Point tR)
	{
		<for you to complete 4 points>
	}
	public double perimeter()
	{
		<for you to complete 8 points>
	}
	public double area()
	{
		<for you to complete 8 points>
	}
}
import java.util.ArrayList;
public class Driver
{
	private ArrayList<Quadrilateral> list = new ArrayList<Quadrilateral>();
	public static void main(String[] args)
	{
		Driver app = new Driver();
	}
	public Driver()
	{
		list.add(new Parallelogram("ABCD", new Point(3,4), new Point(0,0),
					new Point(8,0)));
		list.add(new Parallelogram("EFGH", new Point(-3,4), new Point(1,1),
					new Point(10,1)));
		list.add(new Trapezoid("IJKL", new Point(1,3), new Point(0,0),
				new Point(5,0), new Point(3,3)));
		list.add(new Parallelogram("MNOP", new Point(4,9), new Point(3,2),
					new Point(6,2)));
		list.add(new Trapezoid("QRST", new Point(3,5), new Point(2,2),
				new Point(9,2), new Point(6,5)));
		System.out.println("Label\tPerimeter\tArea\n");

		<for you to complete 4 points>
	}
}
```

Java Concepts Review Assignment:

Pages 489 – 491 Exercises: R10.3 – R10.8, R10.10, R10.11