AP Computer Science — Project 2 - Geometry
Dr. Paul Bailey — Wednesday, August 16, 2017

The purpose of this project is to begin to use Java variables and methods. There are four kinds of variables: static, instance, local, and parameter. We explore each of these, first using only the main method, then using methods as functions.

Create a new directory `Geometry` to hold all of the source files for this project.

Program 1. Type, compile, and run the program `FirstCircle`.

```
public class FirstCircle
{
    private static float PI = 3.14159f;

    public static void main(String[] args)
    {
        float r = 3;
        float C = 2 * PI * r;
        float A = PI * r * r;

        System.out.println("Circle Data");
        System.out.println("Radius: " + r);
        System.out.println("Circumference: " + C);
        System.out.println("Area: " + A);
    }
}
```

In this program, `2`, `3`, and `3.14159f` are constants; the `f` at the end of `3.14159f` tells the compiler that we are using a floating point constant.

The variables `r`, `C`, and `A` are local variables.

The variable `PI` is a static (class) variable.

Program 2. Copy this program into `FirstSphere.java`. Modify the program to output sphere data, including the volume and surface area.

Program 3. Copy this program into `FirstCone.java`. Modify the program to output cone data, including the volume and surface area.

Program 4. Copy `FirstCircle.java` into a new program `SecondCircle`. Refactor this program so that the computational methods are separate from the calling `main` method.

```
public class SecondCircle
{
    private static float PI = 3.14159f;

    public static float circumference(float r)
    {
        return 2 * PI * r;
    }

    public static float area(float r)
    {
        return PI * r * r;
    }

    public static void main(String[] args)
    {
        float D = 6;
        float C = circumference(D / 2);
        float A = area(D / 2);

        System.out.println("Diameter: " + D);
        System.out.println("Circumference: " + C);
        System.out.println("Area: " + A);
    }
}
```

This separates the actual logic for the computations from the code which tests these computations.

The variable `r` is a parameter for the circumference and for the area methods. NOTE: the variable `r` in the `circumference` method *is a different variable* from the variable `r` in the `area` method.

Program 5. Create the class `SecondSphere` in an analogous manner.

Program 6. Create the class `SecondCone` in an analogous manner.

We are now ready to see the final kind of variable, the instance variable. These occur in objects which are instantiated in a separate program.

Program 7. Create the class `Circle`, stored in the source file `Circle.java`.

```
public class Circle
{
    private static float PI = 3.14159f;

    private float r = 0;

    public Circle()
    {}

    public Circle(float r)
    {
        this.r = r;
    }

    public float radius()
    {
        return r;
    }

    public float circumference()
    {
        return 2 * PI * r;
    }

    public float area()
    {
        return PI * r * r;
    }
}
```

Notice that this class does NOT contain a `main` method. It is not intended to be run; it is intended to be used by another program. Type and compile this program.

Program 8. In order to test our `Circle` class, we create a testing program, sometimes known as a "test harness". Following the convention of the programming language C#, we call this class `Program`, and store it in the source file `Program.java`.

```
public class Program
{
    public static void main(String[] args)
    {
        float D = 6;
        float h = 5;
        testCircle(D/2);
        testSphere(D/2);
        testCone(D/2,h);
    }

    public static void testCircle(float r)
    {
        Circle circle = new Circle(r);

        System.out.println("Circle: " + circle);
        System.out.println("Radius: " + circle.radius());
        System.out.println("Circumference: " + circle.circumference());
        System.out.println("Area: " + circle.area());
    }

    public static void testSphere(float r)
    { }

    public static void testCone(float r, float h)
    { }
}
```

Type, compile, and run this program.

Program 9. Create an analogous `Sphere` class. Modify the `Program` class to test the `Sphere` class. Type, compile, and run this program.

Program 10. Create an analogous `Cone` class. Modify the `Program` class to test the `Cone` class. Type, compile, and run this program.