

## Assignment 7 (Due: 27 March, before noon)

Explain the following terms with examples:  
*class methods, classvariables, override, overload.*

(7.31)[Optional, you do not need to submit]

An angle is measured normally in either degrees or in radians where 360 degrees equal  $2\pi$  radians. Write **double** methods `degreesToRadians()` and `radiansToDegrees()` that each expect a single **double** parameter.

Method `degreesToRadians()` treats its parameter as a value in degrees and returns the equivalent number of radians.

Method `radiansToDegrees()` treats its parameter as a value in radians and returns the equivalent number of degrees.

(You must design the methods to be class methods)

For the following questions, you shouldn't get the results by running the program, you should be able to tell the results by just looking at the code segments, and also can explain why.

For Exercises 7.42-7.44, consider the following class `Overload.java`.

```
public class Overload{
    public static void f(int a, double b){
        System.out.println("f(int, double): a = " + a);
        System.out.println("f(int, double): b = " + b);
    }
    public static void f(int a, int b){
        System.out.println("f(int, int): a = " + a);
        System.out.println("f(int, int): b = " + b);
    }
    public static void f(double a, double b){
        System.out.println("f(double, double): a = " + a);
        System.out.println("f(double, double): b = " + b);
    }
}
```

(7.42) What is the output (if any) by the following program? Explain.

```
public class OverloadDemo1{
    public static void main(String[] args) {
        int i = 1;
        int j = 2;
        double x = 3.5;
        double y = 10.2;

        Overload.f(i, j);
        Overload.f(i, y);
        Overload.f(y, x);
    }
}
```

(7.43) What is the output (if any) by the following program? Explain.

```
public class OverloadDemo2{
    public static void main(String[] args) {
        Overload.f(1, 2.3);
        Overload.f(2, 4);
        Overload.f(2.6, 10.5);
    }
}
```

(7.44) What is the output (if any) by the following program? Explain.

```
public class OverloadDemo3{
    public static void main(String[] args) {
        Overload.f(1, 2.3);
        Overload.f(2.3, 4);
        Overload.f(2.6, 10.5);
    }
}
```

For Exercises 7.50-7.53, the following definition is in effect.

```
class A{
    private int count = 0;
    public A(int n){
        count = n;
    }
    public static void d(){
        count = 5;
    }
    public void e(int n){
        { count = n; }
    }
    public void f(int n){
        { int count = n; }
        ++count;
    }
    public static void g(int n){
        count = n;
    }
    public static void h(int n){
        count = n;
    }
}
```

(7.50) Which of the methods in A that are in principle allowed to modify the instance variable count? Why?

(7.51) Which of the methods in A, if any, legally modify the instance variable count? Why?

(7.52) Which of the methods in A, if any, illegally attempt to modify the instance variable count? Why?

(7.53) Which of the methods in A, if any, do not make any attempt to modify the instance variable count? Why?

(7.55) Suppose class C has definition

```
public class C{
    private int data1;
    private int data2;
    private Point data3;
    private Point data4 = new Point();
    public C(){
        data2 = 1;
    }
    public int getData1(){
        return data1;
    }
    public int getData2(){
        return data2;
    }
    public Point getData3(){
        return data3;
    }
    public Point getData4(){
        return data4.toString();
    }
}
```

What is the output of the following program? Why?

```
public class CDemo{
    public static void main(String[] args){
        C c = new C();
        System.out.println(c.getData1());
        System.out.println(c.getData2());
        System.out.println(c.getData3());
        System.out.println(c.getData4());
    }
}
```

For Exercises 7.57 and 7.58, the following definition is in effect

```
public class E{
    private int data = 0;
    public E(){
        // no body used
    }
    public void f(){
        data = 1;
        // f comment
    }
    public void g(){
        data = 2;
        // g comment
    }
}
```

(7.57) If the comment in method f() is replaced with the following statement will class E successfully compile? Why?

```
g();
```

(7.58) If the comment in method g() is replaced with the following statement will class E successfully compile? Why?

```
f();
```