

## CSCI 201.001 Exam 3 Fall 2007 Solution

16 October, 2007

### Problem 1 (30 points)

Declare a reference variable `prices` for an array of Java doubles.

```
double[] prices ;
```

Write a statement to initialize the variable `prices` to reference an array of 15 elements.

```
prices = new double[15] ;
```

Write a Java statement to set the 7th element of the array `prices` to 128.

```
prices[7] = 128.0 ;  
or depending on your interpretation of 7th  
prices[6] = 128.0 ;
```

Write a `for` loop to set a double variable `S` to the sum of the elements in `prices` that are less than 100. [If there are none, set `S` to 0.]

```
double S = 0.0 ;  
for (int i = 0 ; i<S.length; ++i) {  
    if (prices[i] < 100.0) {  
        S = S + prices[i] ;  
    }  
}  
or  
double S = 0.0 ;  
for (int V : prices) {  
    if (V < 100.0) {  
        S = S + V ;  
    }  
}
```

## Problem 2 (30 points)

Consider the following outline of a class with some significant blank lines. This class should contain two constructors, two accessor methods, and two mutator methods. Fill in each of these blank lines to complete these constructors and methods.

```
class Succotash {
    private int limaBeans ;
    private int corn ;

    public Succotash() {
        limaBeans = 1 ;    /* Any old values */
        corn = 1 ;        /* will do. */
    }

    public Succotash(int beans, int corn) {
        limaBeans = beans ;
        this.corn = corn ;
    }

    public int getLimaBeans() {
        return limaBeans ;
    }

    public void setLimaBeans(int beans) {
        limaBeans = beans ;
    }

    public int getCorn() {
        return corn ;
    }

    public void setCorn(int corn) {
        this.corn = corn ;
    }
}
```

Complete a additional Succotash method called `doubleBeans` that doubles the amount of lima beans.

```
    public void doubleBeans() {
        limaBeans = 2*limaBeans ;
    }
```

### Problem 3 (40 points)

The following rather pointless class will be used in this question

```
class Cc {
    private int V1 ;
    private static int V2 = 200 ;

    public Cc() {
        V1 = 100 ;
    }

    public Cc(int A) {
        V1 = A ;
    }

    public int M1() {
        return V1 ;
    }

    public static int M2() {
        return V2 ;
    }

    public void M3(int A) {
        V1 = V2 - A ;
    }

    public void M3(double A) {
        V1 = V2 + (int)A ;
    }

    public int M4(int V2) {
        int T = V2 ;
        V2 = V1 ;
        return T ;
    }

    public int M5() {
        return V1 + V2 ;
    }
}
```

Here's an example of using one of the Cc constructors to initialize a variable X of type Cc.

```
Cc X = new Cc() ;
```

Give an example of using the *other* Cc constructor to initialize a variable Y of type Cc.

```
Cc Y = new Cc(100) ;
```

Assuming that X is a Cc variable (as shown above), which of the following is a legal Java statement. Mark "legal" or "not legal" in each box. Add explanations if you wish.

<pre>int i = X.M1 ;</pre> <p><b>illegal</b></p>	<pre>int i = X.M1 () ;</pre> <p><b>legal</b></p>	<pre>int i = Cc.M1 () ;</pre> <p><b>illegal</b></p>
<pre>int i = X.M2 ;</pre> <p><b>illegal</b></p>	<pre>int i = X.M2 () ;</pre> <p><b>legal</b></p>	<pre>int i = Cc.M2 () ;</pre> <p><b>legal</b></p>

What is printed when the following sequence of statements are executed. It would be easiest if you just wrote the printed value to the right of the appropriate System.out.println statement.

```
Cc X = new Cc() ;           In Cc, V2 is 200  
Cc Y = new Cc() ;         In X, V1 is 100  
System.out.println(X.M5()) ;      In Y, V1 is 100           300  
System.out.println(Y.M5()) ;      300  
  
X.M3(10) ;                 In X, V1 is 190 (V2-10)  
System.out.println(X.M5()) ;      390  
System.out.println(Y.M5()) ;      300  
  
X.M3(20.0) ;               In X, V1 is 220 (V2+10)  
System.out.println(X.M5()) ;      420  
System.out.println(Y.M5()) ;      300  
  
X = Y ;                    X now also refers to Y  
System.out.println(X.M5()) ;      300  
System.out.println(Y.M5()) ;      300  
  
Y.M4(40) ;                 Though 40 is returned  
                                Nothing changes  
System.out.println(X.M5()) ;      300  
System.out.println(Y.M5()) ;      300
```