

CSCI 201.002 Final Exam Fall 2008 Answers

15 December, 2008

Problem 1 (40 points)

In the table below, which continues on the following page, there are thirty expressions in the leftmost column. The value of each of these expressions is either an `int`, `double`, `boolean`, or `String`. In the rightmost column, write the value of each expression. You should write your value in a *simple form* that *clearly indicates its type*. For example, `6.0`, rather than `6` or `2.0*3.0`, is an appropriate value for the expression `2.0*3.0/1.0`, and `"ab"`, rather than `ab` (without the quotes), is an appropriate value for the expression `"a" + "b"`. If you are a bit unsure about your answer and seek partial credit, you may want to include an explanation of your reasoning along with your value.

<i>Expression</i>	<i>Value</i>
<code>"CSCI" + 3 + 4</code>	<code>"CSCI34"</code>
<code>3 + "CSCI" + 4</code>	<code>"3CSCI4"</code>
<code>3 + 4 + "CSCI"</code>	<code>"7CSCI"</code>
<code>0 * (10.0/5.0)</code>	<code>0.0</code>
<code>1/33333</code>	<code>0</code>
<code>10 / 2 * 10</code>	<code>50</code>
<code>(10 / 2) * 10</code>	<code>50</code>
<code>- (6 * 7)</code>	<code>-42</code>
<code>(int) 7.9 < 7.2</code>	<code>true</code>
<code>2 + (1 - 1.0)</code>	<code>2.0</code>
<code>(int) 1.9 + 0.2</code>	<code>1.2</code>
<code>1.9 + 0.2</code>	<code>2.1</code>
<code>5 + 0.1 <= 5</code>	<code>false</code>
<code>(double)5/2</code>	<code>2.5</code>
<code>(double)(5/2)</code>	<code>2.0</code>
<code>(int)5.0/2.5</code>	<code>2.0</code>
<code>(int)(5.0/2.5)</code>	<code>2</code>
<code>(1.0 + 1) + "x"</code>	<code>"2.0x"</code>
<code>1.0 + (1 + "x")</code>	<code>"1.01x"</code>
<code>Math.PI - Math.PI</code>	<code>0.0</code>
<code>1e4/1000</code>	<code>10.0</code>

5 < 4 4 < 3	false
5 < 4 && 4 < 3	false
5 < 4 == 4 < 3	true
! (5 < 6)	false
"be " + (5<6)	"be true"
7 / 2	3
7 % 2	1
201 % 1500	201
1e30 - 1e30	0.0

Problem 2 (4 points)

In the sequence of Java code shown below, Picture, Pixel, and Sound objects called qzPicture, qzPixel, and qzSound are created

```
Picture qzPicture = new Picture("Pisgah.jpg") ;
Pixel    qzPixel   = new Pixel(qzPicture, 201, 201) ;
Sound    qzSound   = new Sound("Braes.wav") ;
```

In the following table are some method descriptions taken from the documentation of the Picture, Pixel, and Sound classes.

From the Picture class

int	getWidth() Method to get the width of the picture in pixels
static void	setMediaPath(String directory) Method to set the media path by setting the directory to use

From the Pixel class

void	setGreen(inv value) Method to set the green to a new green value
------	---

From the Sound class

static void	convert(String mp3File, String wavFile) Method to convert an mp3 sound into a wav found
boolean	isStereo() Method to check is sound is stereo or not

Write two different Java statements below. One should invoke a static method and the other should invoke a dynamic (non-static) method chosen from the five methods shown above.

```
// Static
Picture.setMediaPath("/home/brock/csci/201") ;
// Dynamic
qzSound.isStereo() ;
```

Problem 3 (3 points)

In the spirit of the **Using the Math class** lab, complete the method `plotQuiz` shown below so that it returns $x^2 + x + 1$.

```
static double plotQuiz(double x) {  
    return x*x + x + 1 ;  
}
```

Problem 4 (7 points)

In the spirit of the **Array operations** lab, complete the method `operQuiz` shown below so that it returns the smallest element of the array `v`. You may assume `v` has at least one element.

```
static int operQuiz(int v[]) {  
    int smallest = v[0] ;  
    for (int x : v) {  
        if (x < smallest) {  
            smallest = x ;  
        }  
    }  
    return smallest ;  
}
```

Problem 5 (4 points)

Continuing in the spirit of the **Array operations** lab, complete the method `modQuiz` shown below so that it adds 100 to each element of the integer array `v`.

```
static void modQuiz(int v[]) {  
    for (int i=0; i<v.length; ++i) {  
        v[i] = v[i] + 100 ;  
    }  
}
```

In the problems of the next two pages, you are going to write parts of a Java class that represents simplified cell phone calling “plans”. For each plan your class should keep up with only three pieces of information:

- 1)The monthly allowance of minutes ;
- 2)The monthly fee ; and
- 3)The per-minute fee (for when you exceed the monthly allowance of minutes).

Problem 6 (2 points)

Assuming that your class will be called `PhonePlan` and it will need no Java `import` statements, what should the first line of your class be?

```
public class PhonePlan
```

Problem 7 (1 points)

What is the name of the file in which your program should be stored? In your answer, ignore the directory part of the file name.

```
PhonePlan.java
```

Problem 8 (3 points)

Write three Java declarations for the three fields (instance variables) needed to store the monthly allowance of minutes, the month fee, and the per-minute fee. Assume that the minute allowance is stored in a Java `int` and the fees are stored as Java `doubles`.

```
// Your names may be different  
int monthlyAllowance ;  
double monthlyFee ;  
double perMinuteFee ;
```

Problem 9 (8 points)

Write three modifier (`set`) methods for your class: One for each of your fields. (Yes, it is tedious without cut-and-paste, but all three methods *can* be used on the next page.)

```
// Your names may be different  
public void setMonthlyAllowance(int allowance) {  
    this.monthlyAllowance = allowance;  
}  
  
public void setMonthlyFee(double monthlyFee) {  
    this.monthlyFee = monthlyFee ;  
}  
  
public void setMinuteFee(double minuteFee) {  
    this.perMinuteFee = minuteFee;  
}
```

Problem 10 (3 points)

Write an accessor (get) method for the per-minute fee.

```
public double getMinuteFee() {
    return perMinuteFee ;
}
```

Problem 11 (4 points)

Complete the following constructor for your class.

```
public PhonePlan(int minutes,double monthlyFee,double minuteFee)
{
    // Be consistent with names used in Problem 9
    setMonthlyAllowance(minutes) ;
    setMonthlyFee(monthlyFee) ;
    setMinuteFee(minuteFee) ;
}
```

Problem 12 (2 points)

Write a no-arg constructor for your class. You'll need to provide reasonable defaults. It really should fit within the allocated space.

```
public PhonePlan() {
    this(400, 49.95, 0.35) ;
}
```

Problem 13 (5 points)

Finally write a method that is passed the number of minutes used in a month and returns the cost of the plan for that month. Let's hope you don't exceed the allowance!

```
public double chargeForMonth(int minutesUsed) {
    // Be consistent with problem 8 variable names
    double overTheLimit = 0.0 ;
    if (minutesUsed > monthlyAllowance) {
        overTheLimit = minutesUsed - monthlyAllowance ;
    }
    return monthlyFee + preMinuteFee * overTheLimit ;
}
```

The questions on this page are concerned with computing average of positive numbers stored within an integer array, where some of the numbers are missing. (Think of homework scores where some homeworks haven't been graded and the ungraded homeworks are indicated with negative "scores".) Consider the array shown in the following picture:

70	80	100	-1	90	30	100	100	-1	100
----	----	-----	----	----	----	-----	-----	----	-----

The array itself has ten elements, but two of them are negative numbers indicating missing values. When computing the average of the array, the positive numbers should be added (giving 670) and divided by the number of positive elements in the array (8) which yields an average of 82.75. In this problem assume the array, which we'll call `scores`, has been declared as follows:

```
int[] scores ;
```

Also assume that the array was later created and given a set of values.

Problem 14 (10 points)

Write either a single `for`-loop or two separate `for`-loops that store into the variable `numberScores`, the number of elements of the array `scores` that have a positive value, and into the variable `totalScores`, the result of adding the positive values within the array `scores`.

```
int numberScores = 0 ;
int totalScores  = 0 ;
for (int possibleScore : scores) {
    if (possibleScore >= 0) {
        numberScores = numberScores + 1 ;
        totalScores  = totalScores+possibleScore ;
    }
}
```

Problem 15 (4 points)

Finally, write a piece of code that uses `numberScores` and `totalScores` and sets the variable `averageScore` as follows:

1)If `scores` has no positive elements, `averageScore` is set to `-1.0`.

2)Otherwise, `averageScore` is set to the average of the positive elements of `scores`.

```
double averageScore = -1.0 ;
if (numberScores > 0) {
    averageScore
        = (double)totalScores / numberScores ;
}
```