UNCA CSCI 202
Middle Exam 2013
9 April, 2013

This is a closed book and closed notes exam. It is a 100-minute exam to be given within the regular class meeting time. Calculators, PDA's, cell phones, and any other electronic or communication devices may not be used during this exam.

Name: _____

If you want partial credit for imperfect answers, explain the reason for your answer!

Problem 1 (9 points)
Complete the `manyMe` method so that it creates and returns a `String` array of 2013 elements. Your name should be stored in all 2013 elements of the array.

```
public static String[] manyMe() {




}
```

Problem 2 (6 points)
Suppose there is a class `Match4Exam` with methods with the following headers:

```
public        int    fancyOpA (int x, String y) ;
public        double fancyOpB (int x, double y) ;
public static double fancyOpC (int x, int y) ;
```

and that `fancyX` and `fancyY` are objects of type `Match4Exam`.

Write Java statements to do each of the following:

Invoke `fancyOpA`
Invoke `fancyOpB`
Invoke `fancyOpC`

Problem 3 (15 points)
The table below contains two columns. The leftmost column is a Java expression. In the rightmost column write the value of the expression as a Java literal. If the value is a `double` or a `boolean` or a `String` be sure to use the Java syntax for writing `double`s or `boolean`s or `String`s. Some of these are tricky, but none require complex arithmetic.

`202 / 10`	
`202 % 10`	
`3 * 5 / 10`	
`3 * (5 / 10)`	
`3.0 * 5 / 10`	
`3.0 * (5 / 10)`	
`0 * Math.PI * Math.E`	
`"" + 202`	
`"CSCI" + 202`	
`"CAT" + "MOUSE"`	
`5 < 6.0`	
`! (5 == 5)`	
`5 == 5 && 6 == 6`	
`5 == 5 || 6 == 6`	
`5 > 6 ? 5 : 6`	

Problem 3 (8 points)
Circle all the Java tokens in the following assignment statement:

```
s = "CAT" + 3.5 + (int) x ;
```

Now circle all the (sub)expressions in the following arithmetic expression:

```
"CAT" + 3.5 + (int) x
```

Problem 4 (16 points)

Suppose you are asked to write a Java class within the `edu.unca.nm.nm222` package. What would be the first line of Java within your program?

If your program needs to import the `java.net.Socket` class, what would be the second line of Java within your program?

If your program is to implement a Java class called `Problem4` that could be accessed from classes outside of the `edu.unca.nm.nm222` package, what would be the Java class header for your program?

What is name the file in which you must store your `Problem4` class? You do not need to give the directory part of the name, just the part after the slash.

How would you declare a member variable of type `double` called `point` within your `Problem4` class so that `point` can be accessed ***only*** by methods of the `Problem4` class?

Write an accessor ("getter") method for the `point` member that can be invoked from outside of the `Problem4` class.

Write a mutator ("setter") method for the `point` member that can be invoked from outside of the `Problem4` class.

If `p` is an object of type `Problem4`, write a very short section of Java code that uses your accessor and mutator methods to triple the number stored in the `point` member of `p`.

Problem 5 (10 points)
Write a `static` ***recursive*** method `numZeros` to compute the number of times that 10 can be divided into a strictly (greater than 0) positive integer *n* by using the following recursive formula:

`numZeros`(*n*) = `numZeros`(*n*/10) + 1, if *n*%10 is 0
`numZeros`(*n*) = 0, otherwise

Do not think too hard about this one. Just implement the two conditions. Four or five lines of Java should do the trick.

Start with the following method header

```
public static int numZeros(int n) {
```

Don't tell anyone about all the unused space on this page.

Problem 6 (12 points)
Write a static method called `addEmUp` that receives an array of integers as its single argument and returns the result of adding all the elements of that array. If the array contains no elements, your method should return 0.

It is up to you to provide an appropriate method header.

This method can be written in about eight lines of Java.

Problem 7 (12 points)
Assume a class `Point1080` has been defined with a constructor that takes two integers as arguments (as was the case in many labs). Write a section of Java code that

1) Creates a 202×202 two-dimensional array called `pts` of `Point1080` objects.
2) Initializes the array by setting `pts[`*i*`][`*j*`]` to the object created by invoking the `Point1080` constructor with *i* and *j*.

About eight lines of Java are needed for this answer.

Problem 8 (12 points)
You don't need to know anything about the `java.net.inetSocketAddress` class other than what you are told below to answer this question. (Think about how you used the `java.net.URI` class in Homework 4.) You should also assume that the appropriate import statements for `java.net.inetSocketAddress` have been provided for your code.

The `java.net.inetSocketAddress` class defines one constructor as follows:

```
public inetSocketAddress(String hostname, int port)
```

This constructor can throw the exception `IllegalArgumentException`.

The `inetSocketAddress` class also contains the following method:

```
public String getHostName()
```

Write a section of Java code that creates an `inetSocketAddress` object named `sockAddr` using `"www.nm.unca.edu"` and 8080 as arguments to the constructor described above.

If `IllegalArgumentException` is thrown by the constructor, your Java code *must* print a useful error message. If no exception is thrown by the constructor, your Java code *must* invoke the `getHostName` method to print the host name for `sockAddr`. (In your answer, do not just assume that `getHostName` will not return the first argument passed to the constructor. It rarely does.)

Again, about eight lines of Java are needed for this answer.