

**CSE 11**  
**Fall 2013**  
**Program/Homework Assignment #3 (100 points)**

**Due: 21 October 2013, 11:00pm**  
Covers Chapters: 5 - 8

This is a predominantly homework assignment with a number of small programs and some questions closer to the format of an exam. Most of your answers will be turned in as a single pdf (Portable Document Format), You can use OpenOffice on the lab machines or in the class VM to export a PDF. Call this file PR3.pdf. Include your name inside of this file.

Also create a file PR3.txt that has just the following lines with your name, login and ID filled in as in previous programs.

LOGIN: <your login id>  
NAME: <your first and last name>  
ID: <your student id>

Turning in your homework. As in previous assignments we'll use a bundle program.

Run `/home/linux/ieng6/cs11e/public/bin/bundleP3` in the directory that contains your homework files. It will look for the following files

PR3.txt  
PR3.pdf  
RandomCircles.java

Note that in all commands, the "\$" indicates the shell prompt. You do not type in the \$, only what comes afterwards. You hit "enter" or "return" to execute a command in the shell

Please note that: JAEA = "Java, An Eventful Approach". A problem as JAEA: Exercise 1.5.2 means do exercise 1.5.2 in your textbook.

Include code – means include a printed copy of your code.  
Include output – means turn in screen grab of the output of your code.

**Problem #1: (5 points)**

What is the difference between an accessor method and a mutator method for a class?

**Problem #2 (5 points)**

Go to the book website  
<http://eventfuljava.cs.williams.edu/library/objectdrawJavadocV1.1.2/index.html>

Looking at the documentation for the objectdraw library, How many different ways are there to construct a `FramedRect` object? What are the signatures of the constructor(s).

If you have a `Text` instance, what method is called to change the font size of the `String` to be displayed?

If you have `VisibleImage` instance, what method would you call to determine if the image should be displayed on the canvas?

### Problem #3 (5 points)

Read section 5.5.2 of your book. It describes using `Math.round()` to convert doubles to integers (Java will not do this type conversion automatically). Look at <http://docs.oracle.com/javase/6/docs/api/java/lang/Math.html> to see a full description of methods in the `Math` class. What do `ceil()` and `floor()` do? These methods return doubles, If you want an integer (long) version of these two functions, just using the `Math` class functions, how would you accomplish that? In other words, in the following code segment how would you compute `low` and a `high` so that they hold the floor and ceiling of `x`, respectively?

```
double x = 10.35;
long low, high;
```

### Problem #4 (5 points)

Download `Key.java`, `Lock.java`, `SampleKey.java` from Lecture 7. Compile and run them.

Now change the line in `Key.java`

```
myLock = theLock;
```

to

```
myLock = new Lock();
```

recompile `Key.java` then and re-run `SampleKey`. What does the unmodified program print out? What does the modified program print out. Explain why the output is different after the one line code change?

### Problem #5 (10 points)

Gathering arguments from the command line. in the following, we'll use a very primitive way to gather arguments from the command line and make them usable variables in our programs. Later, we'll learn more compact ways to read arguments. Type in, compile and run the following program

```
public class ReadArgs
{
    static public void main(String [] args )
    {
        int numargs = args.length; // # arguments typed in
        int argindex = 0;
        int argvalue;
        while ( argindex < numargs )
        {
            argvalue = Integer.parseInt(args[argindex]);
            System.out.println("Argument " + argindex +
                               ":" + argvalue);
            argindex++;
        }
    }
}
```

To see what the program does, run with  
`java ReadArgs 20 200 50`  
and then  
`java ReadArgs 10 20`

Note that `Integer.parseInt()` is defined by Java to convert a `String` into an integer.

There exists a similar set of options for the `Double` class ( for example, <http://docs.oracle.com/javase/6/docs/api/java/lang/Double.html>, and <http://docs.oracle.com/javase/6/docs/api/java/lang/Integer.html> ). Note: a `Double` object type is different than a `double` primitive type.

`args[ argindex ]` retrieves one element of the `args` array (We haven't done arrays yet). `args[ 0 ]` is the first argument typed in on the command, line, `args[ 1 ]` is the second one. Finally, `args.length` tells us how many arguments were typed in.

Modify the code to

1. read **no more than 3** arguments and call them `arg1`, `arg2`, `arg3`. These should each be of type `int`
2. instead of printing the arguments, print the product of the arguments ( only calculate the the product of the first 3 arguments in the case where more than 3 were typed in at the command line). If less than three arguments are typed in, calculate the product of the arguments given to you.
3. should print nothing if no arguments are listed

Include code

Include output of your modified program when run with 0,1,2,3, and 4 arguments given on the command line

### problem #6 (10 points)

T	F	All variables are initialized to zero by java
T	F	<code>void sign(double x);</code> and <code>void sign(double y);</code> have identical signatures
T	F	Constants must be declared public
T	F	A static variable is only visible with the class that defines it
T	F	It is legal use the <code>private</code> access modifier to declare a temporary variable within the statement block that defines a method
T	F	If a java source file defines a public class called <code>BigClass</code> , the file must be called <code>BigClass.java</code>
T	F	It is legal to perform an assignment within a boolean expression
T	F	Variables of type <code>int</code> are automatically converted to doubles, when needed.
T	F	Strings are a primitive type.
T	F	A class can only define one constructor

**Problem #7 (5 points)**

JAEA: Exercise 6.8.4

**problem #8 (5 points)**

JAEA: Exercise 7.1.2

**problem #9 (10 points)**

Modify the program you created in problem 5 and call it RandomCircles to process two command line arguments. The first is the number of random circles to draw, the second is the size of the canvas to create. For example, if you wanted to create 12 random circles on a 400x400 canvas, you would run from the command line as

```
$ java RandomCircles 12 400
```

For many commands we would specify this as

```
RandomCircle <ncircles> <canvas_size>
```

where <ncircles> is a required command-line parameter. and <canvas\_size> is a required command line parameter

RandomCircles should behave as follows:

1. it should create a canvas of the size <canvas\_size>x<canvas\_size>
2. it should create <ncircles> randomly placed circles on the canvas
3. The “diameter” of each circle should 1/10th of the canvas\_size parameter
4. each circle should have a different, randomly-generated color

You may use either FilledOval or FramedOval to create your circles

Verify your code works properly with

```
$ java RandomCircles 10 400
```

```
$ java RandomCircles 17 600
```

bundleP3 will look for RandomCircles.java

**problem #10 (5 points)**

Use DeMorgan's Laws to rewrite the following boolean expressions

```
(x == 25 || y < 45)
```

```
(delta < epsilon || i > 1000)
```

```
(charge < threshold && !hot)
```

```
(apple.color == Color.RED && leaf.color != Color.YELLOW)
```

```
(l.getEnd().getX() < 50 || time >= MAX_TIME)
```

**problem #11 (5 points)**

JAEA: Exercise 7.7.3

**problem #12 (10 points)**

Using the NestedLoopGray.java program presented in lecture as a starting point, do JAEA 7.11.1. (This program is available from the class website)

Include code  
Include output

**problem #13 (5 points)**

describe the following:

1. The difference between a `public` and `private` class variable
2. The difference between a `private` class variable and `private static` class variable
3. The difference between a `private` class variable and a local variable

**problem #14 (5 points)**

write a public method called `sumSquares` that returns an integer. `SumSquares` has two integer arguments called `low` and `high`. `SumSquares` computes the sum of the squares of the integers in the range `[low,high]`. Example: `sumSquares (1,4)` should return  $1^2 + 2^2 + 3^2 + 4^2 = 30$ . You only need to write the method.

**problem #15 (5 points)**

JAEA: Exercise 8.6.1

**problem #16 (5 points)**

JAEA: Exercise 8.6.2

**The following are recommended Exercises**

**Exercise #1**

Getting more familiar with the Unix shell:

1. What command is used to create a directory?
2. What command is used to change into that directory?
3. What is the command to display the current working directory?
4. What do `'..'` and `'.'` mean
5. How do you list the contents of the current directory? Suppose I wanted to list the contents of the system directory `"/usr/bin"`, what is the complete command?
6. What is the command to rename a file? give an example of renaming the file `"MyFile.java"` to `"YourFile.java"`

The program `"man"` is the unix command (on the lab machines and in the class virtual machine) for describing how commands are to be called.

type in at the command line `$ man uptime`

Based on that output, what does this command do? In unix, command-line parameters that are enclosed in square brackets `[]` are optional. What is the optional argument to `uptime`?

We call the output of `"man <command>"` the `"man-page for the <command>"`. That is

uptime's man-page is the output of “man uptime”

## Exercise #2

Unix environment variables. Unix has environment variables the help programs figure out where things are. One of the important variables is called `PATH`. This is a list of directories (or paths) in the file system where the system searches for executables. Another program called `which` can tell you the full path name to particular program. For example `$ which java`, will print out the full path to the java executable

On a lab machine (or your desktop or the lab virtual machine), what are the full paths to

1. java
2. ls
3. wc

Look at the man-page for java and javac (there are **many** options). In particular look at the one option that we've used in our programs called `-cp` (or `-classpath`, they are the identical option). If you have a `CLASSPATH` environment variable and you use `-classpath` to either java or javac, which one is used?

to set the `CLASSPATH` environment variable to `'*':'. '` use the command

```
$ export CLASSPATH='*':'. '
```

this can save you time typing for future programs!

## Exercise #3

`wc` is the command to count words in a file.

According to the man page what are the three numbers output by default?

If you type in “`wc -l NestedLoopGray.java`” in the directory where the file is located, what is the meaning of the number printed out?

How would you print just the number of words in `NestedLoopGray.java`? How many words are there in the `NestedLoopGray.java` program?