

In this class you will have many programming assignments, but hopefully you'll quickly learn that programming is just one part of what's required. The purpose of Assignment #0 is to acquaint you with all the information, software resources, and technical systems that you will need this semester.

1 Your Assignment

There are several things you need to do to complete and turn in a project, and it's vital that you solve any difficulties now, before you have a real programming assignment to worry about. There will still be a small amount of programming to force you to use java docs and get to know your editor of choice, but this should hopefully not take much time.

For this assignment, you will do the following:

- Find and successfully log into a computer in one of the Painter computer labs. (You may also use your own computer.) Make sure you have a CS account!
- Choose a programming environment / editor that you will use to write and debug your programs. (Including setting up and installing them if using your own computer.) Whatever you like to use is fine, but always make sure your code runs from the command line, because this is how it will be tested!
- Create a new Java project with the included source file and editor of your choice.
- Use java docs for help and finish the program.
- Test your program results and ensure that it runs correctly on the lab's javac.
- Write a pdf (or plain text) report describing your algorithm and any difficulties you encountered with this process.
- Successfully turn in your source file and pdf report using the labs web based turnin program.

None of these tasks should be time consuming, but technical glitches always arise, so don't wait until the last minute!

2 Details

To be a successful programmer, you will need to become adept at finding your own information. In this course, we will not spoon feed you the details, but you will often find useful links and information on the TA's website. Remember that the purpose of this assignment is largely for you to learn how to find out these details by yourself.

2.1 Write Your Program

Rampant use of antibiotics in salad dressing has produced a new strain of bacteria that cannot be destroyed. Scientists need your help to determine how many of the lethal creatures may exist within the next few hours. This simple program is designed to see how many bacteria will remain after a given time. It is assumed that bacteria are immortal and divide at regular intervals (in this example will be hard coded to 30 seconds). The program takes as its argument the number of elapsed hours and then prints the total number of bacteria to stdout.

You must complete the included code (download the file from the assignment page) so that it correctly prints out the exact number of bacteria existing after a certain amount of time. There are many ways to write this code for example, you might try to find a java command that might help you or write your own code.

Please make sure your output format exactly matches the format given (see below).

2.2 Test Your Program

In this class, testing is far more important than the code itself. Code that does not work will receive little credit. Be sure to test your code for various inputs. If you are unsure what inputs to expect, ask.

Here are some sample results that you can use to help test your code.

```
Java BacteriaCount 0.1
(Number of divisions: 12)
After 0.1 hours, there are roughly 4096.0 Bacteria.
There are exactly:
4096
bacteria!

Java BacteriaCount 2.5
(Number of divisions: 300)
After 2.5 hours, there are roughly 2.037035976334486E90 Bacteria.
There are exactly:
20370359763344860862684456884093781610514683936659362506361404493543812997633367061833973
76
bacteria!

Java BacteriaCount 24
(Number of divisions: 2880)
After 24.0 hours, there are roughly Infinity Bacteria.
There are exactly:
92552363180904791850780891501317434300699049797689030749153376097694282541215739638503152
87853722510320306814009016420024778334267671843873439557057958805535553704153249706088691
30154498070284494448923578669897222826668401826976211221365596131277796416266571966724828
94411667397732453700020933213597872100177264586037204573216890444569602185257412805546320
97325832646852540751100434623187335372481104916056975825417268451598867168708939858973565
69465053999819856631633580110898489602220450743708504681825102467021819092022398105605148
03650679084211928933025728350091693715035926547277528354604012230126792011579498858174013
34978145078092263160805655406215533234726566350101568238163819914667308284334573482895318
80490713853927338907650295839195196291190183108782773867294313240177739054272302997307305
371870273048855559958177900851568273553787307555754789395471794176
bacteria!

Java BacteriaCount -1.0
Error - hoursElapsed < 0.0

Java BacteriaCount thirty
Exception in thread "main" java.lang.NumberFormatException: For input string: "thirty"
    at sun.misc.FloatingDecimal.readJavaFormatString(Unknown Source)
    at java.lang.Double.parseDouble(Unknown Source)
    at BacteriaCount.main(BacteriaCount.java:70)
```

2.3 Write Your Report

As in real life, your report will be your primary deliverable and the main document that your audience will read. The report concisely describes what you did and how you did it. For this assignment, you should be brief, but you must include the following: (1) A brief overview of the assignment and your goals in completing it, (2) a description of what you did—the tools you chose, the algorithm you used to complete the program, and the test cases you tried in debugging, and (3) a discussion of anything interesting or surprising or difficult that happened along the way.

The electronic copy of your report should be in plain ASCII text or Adobe PDF format. Other formats, such as Microsoft Word, will not be accepted. This builds character. As is expected for all assignments in this class, *you should clearly acknowledge any code that you use that is not your own.*

2.4 Karma Problems

This assignment is designed to be quick, but if you want an extra challenge, you might:

- Analyze the algorithm you used: How much memory does it require? How fast does it execute?
- Compare more than one algorithm: Implement this in more than one way and compare the results.
- Test your code on more than one platform and/or development environment: Did you need to make any changes? Which environment was easiest?

3 What To Turn In

Hard copies. Print out copies of your report and your code to Gem Naivar. For this assignment only, you can submit your hard copy in discussion section on Tuesday September 4th.

Electronic Copies. Electronically submit a single JAR file (details about JAR files are given below). This JAR file should contain your report and your source code. Your electronic copies should match your hard copies. The electronic version of your report should be in plain ASCII text format or Adobe PDF format. *Please name the .jar file progN.jar*, where N is the number of the assignment. For those not familiar with jar, the basic syntax to create a JAR file is:

```
jar cf filename.jar list-of-files . . .
```

For example, when you finish this assignment and want to turn in BacteriaCount.java and report.txt in the directory where those files are located, you should issue the command:

```
jar cf prog0.jar BacteriaCount.java report.txt
```

The above would create prog0.jar, which can then be turned in.

4 The turnin Program

Use the turnin program that is available on the PC's in the Painter microlab. See the class webpage for more details. Your assignment is due at 4:00pm on the due date. Late assignments will be penalized 10% per day.