

CS1322 Homework 1

Due: Tuesday, January 18 @ 6:00 pm

Introduction

Welcome to CS1322, Introduction to Object Oriented Programming. If you do not understand the title of this course, you will in due time. First though, we must learn how to write simple programs. The computer language of choice in this course is Java, so before running any programs, we must install the Java compiler and the Java Virtual Machine (JVM). Instructions to do so can be found at the following link:

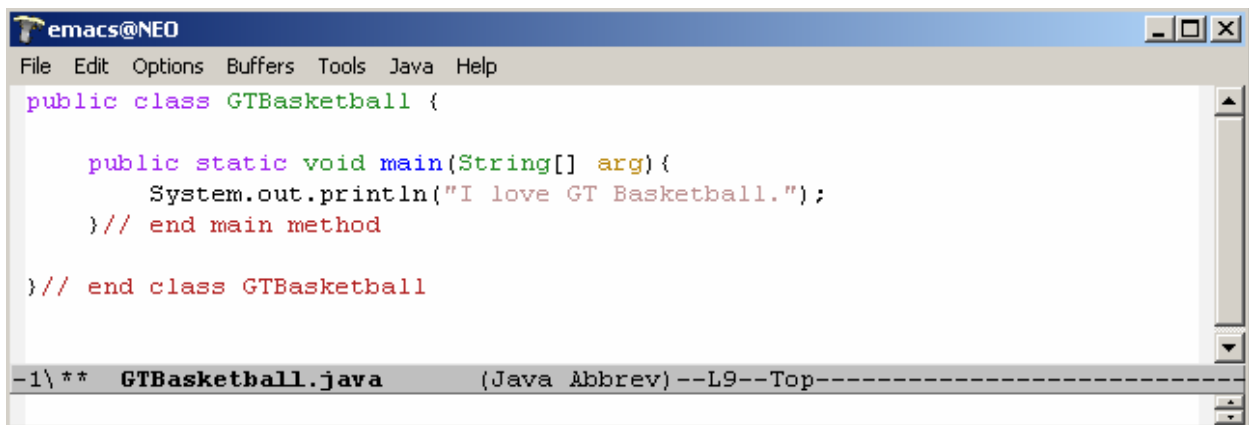
http://www.cc.gatech.edu/classes/AY2005/cs1322_spring/started/index.html

The purpose of the compiler is to translate our human readable code into byte code (computer readable ones and zeros) and in the process check our code for syntactical errors (compiler errors). The JVM allows the byte code to be executed. In this homework, your task will be to write a simple program, compile it, and execute it. You will then purposefully insert errors into the code and record the compiler messages.

Assignment 1.1: Writing a simple program

This assignment will guide you through the process of writing a simple java program. Please take time to understand the syntax of the code because without these fundamentals, we cannot create more complex programs. Also, beware that improper syntax on exams and quizzes will result in lost points.

When writing a program, we must first create a java file. In the editor of your choice (shown here is emacs), open a file and call it GTBasketball.java. In the empty file, type the following code and save it. Please note that indention is also very important because it makes the code readable and easy to debug.

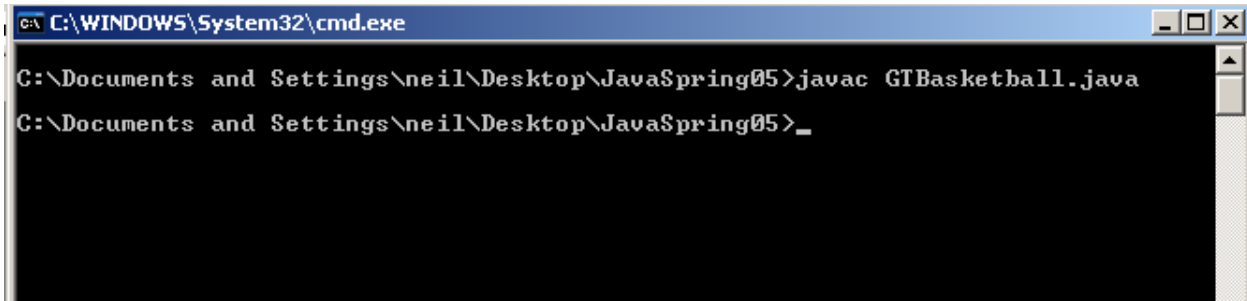


```
emacs@NEO
File Edit Options Buffers Tools Java Help
public class GTBasketball {
    public static void main(String[] arg) {
        System.out.println("I love GT Basketball.");
    } // end main method
} // end class GTBasketball
-1\ ** GTBasketball.java (Java Abbrev) --L9--Top
```

Voila! You have just written your first Java program. Let's go through the code so you can understand what is going on.

Line Number	Code Segment	Explanation
1	<code>public class GTBasketball {</code>	Every java file starts with public class <<filename>> followed by a curly brace. Remember for every beginning curly brace you need an ending curly brace.
2	<code>public static void main(String[] arg) {</code>	The details of this line are not important at the moment, but know 2 things: this segment of code is called a method signature, and this method is a special one called 'main'.
3	<code>System.out.println("I love GT Basketball");</code>	Okay, so now we are going to make our code do something. For this first homework assignment, all we want to do is print something to the console/command prompt/ shell (whatever you want to call it). The 'System.out.println' tells the computer you want to print something to the screen. What you put inside the parentheses in quotes is the text you want to print out. Also note that this line of code ends in a semi-colon.
4	<code>}// end main method</code>	This curly braces matches with the one in line 2, which means anything in between the braces belongs inside the main method. The '// ' after the curly brace allows us to place comments. Anything after a '// ' is not considered an instruction for the computer to execute. It is good practice to label the end of curly braces so we know what they belong to.
5	<code>}// end class GTBasketball</code>	This brace ends the class. Same idea as above.

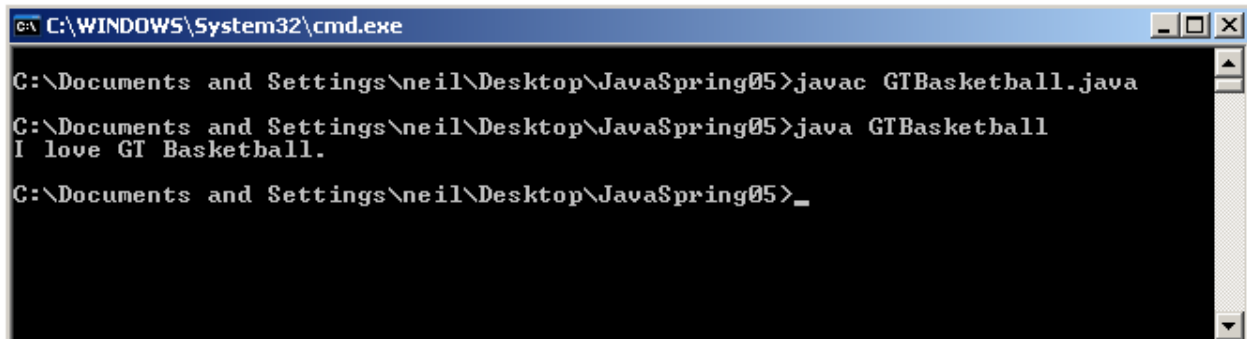
After saving your file, there are several ways to compile your code. Remember compiling will alert you of any errors you might have made. If you are using an IDE like JGrasp, look in the toolbar at the top for a button that says compile. If you want to compile your code from the command prompt, open a command prompt window (in MS Windows, go to Start->Run and type 'cmd' in the box. A black screen should appear with a prompt) and go to the directory in which your file is saved. Type 'javac GTBasketball.java'. If everything goes well, you should see a new prompt appear.



```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\neil\Desktop\JavaSpring05>javac GTBasketball.java
C:\Documents and Settings\neil\Desktop\JavaSpring05>_
```

Congratulations. You have just compiled your first java program. Notice that in the same directory a file called GTBasketball.class has appeared. This is the byte code we mentioned earlier.

Now let's run it and see what happens. If you are using an IDE, find the run button in the toolbar. Otherwise, in the command prompt, type 'java GTBasketball'. You should see something like the following.



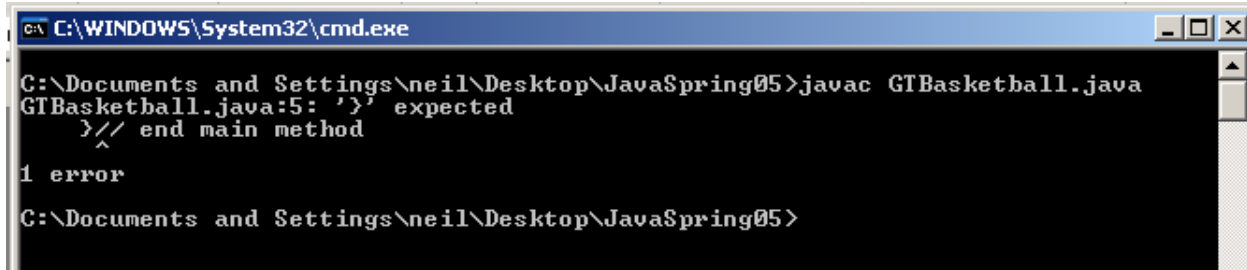
```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\neil\Desktop\JavaSpring05>javac GTBasketball.java
C:\Documents and Settings\neil\Desktop\JavaSpring05>java GTBasketball
I love GT Basketball.
C:\Documents and Settings\neil\Desktop\JavaSpring05>_
```

Again, congratulations. You have just run your first java program. Now, let's see what happens when there are errors in our code.

Assignment 1.2: Introducing Errors into a Program

Introduce the following errors, one at a time to the program from the programming assignment 1.1. Record any error messages that the compiler produces in a file called P1Errors.txt. Fix the previous error each time before you introduce a new one. If no error messages are produced, explain why.

- o Change *GTBasketball* to *GTbasketball*
- o Change love to LOVE
- o Remove the first quotation marking the string
- o Remove the last quotation mark in the string
- o Change main to man
- o Change println to bogus
- o Remove the semicolon at the end of the println statement
- o Remove the last brace in the program



```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\neil\Desktop\JavaSpring05>javac GTBasketball.java
GTBasketball.java:5: ')' expected
    }// end main method
    ^
1 error
C:\Documents and Settings\neil\Desktop\JavaSpring05>
```

Turn-in Procedures

After you have finished the above assignments, turn them via [Webwork](#) You will be submitting two files:

- GTBasketball.java
- P1Errors.txt

A good practice (called “safe submission”) to begin the semester is to download your files after they have been submitted – using the retrieve submissions button in Webwork – to an empty directory, so that you may recompile and test them. Missing or non-compiling files will not be given any credit. Also recall that late homeworks will not be accepted. Remember that although the deadline for the homework is at 6:00 pm, we provide a 6 hour grace period until midnight. If you are able to turn it in during that time, it is OK, but you will not be guaranteed to turn it in then.