

HW01 - Submitted

Title	HW01
Student	Lodhia, Rahmaan Mohammed
Submitted Date	Aug 24, 2009 9:40 am
Grade Scale	Points (max 100.0)

Instructions

CS1331 Homework 1

Due Mon Aug. 24, 2009 11:55 pm

Introduction

Welcome to CS1331, 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/AY2009/cs1331_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 GTBuzz.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.



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 GTBuzz {</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. </filename>
2	<code>public static void main(String[] args) {</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'm a ramblin' wreck.");</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 GTBuzz</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 GTBuzz.java'. If everything goes well, you should see a new prompt appear.

```

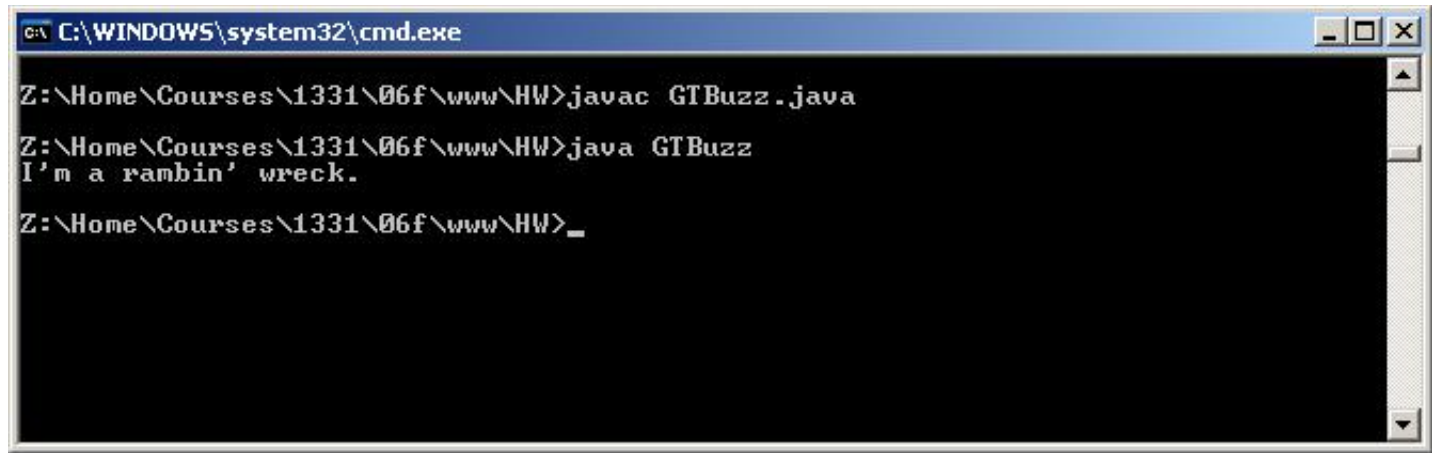
C:\WINDOWS\system32\cmd.exe
Z:\Home\Courses\1331\06f\www\HW>
Z:\Home\Courses\1331\06f\www\HW>javac GTBuzz.java
Z:\Home\Courses\1331\06f\www\HW>_

```



Congratulations. You have just compiled your first java program. Notice that in the same directory a file called GTBuzz.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 GTBuzz'. You should see something like the following.



```
C:\WINDOWS\system32\cmd.exe

Z:\Home\Courses\1331\06f\www\HW>javac GTBuzz.java

Z:\Home\Courses\1331\06f\www\HW>java GTBuzz
I'm a rambin' wreck.

Z:\Home\Courses\1331\06f\www\HW>_
```



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 the compiler produces in a file called P1Errors.txt. If no error messages are produced, explain why. Note that there is a difference between compile-time and run-time errors: if your program compiles but does not run, explain why. **Fix the previous error, if any, each time before you introduce a new one.**

- Change GTBuzz to GTbuzz (note lower case 'b')
- Change wreck to WRECK
- Remove the first quotation marking the string
- Remove the last quotation mark in the string
- Change main to man
- Change println to bogus
- Remove the semicolon at the end of the println statement
- Remove the last brace in the program

Turn-in Procedures

After you have finished the above assignments, turn them in via [T-Square](#). You will be submitting two files:

- GTBuzz.java (undo any errors you introduced, so it should compile and run without errors)
- P1Errors.txt

For this and all future homeworks, you must include a collaboration statement with your homework *even if you worked alone*. Put the collaboration statement at the top of GTBuzz.java as a comment. See

http://www.cc.gatech.edu/classes/AY2009/cs1331_spring/collab.html for details on the collaboration statement.

A good practice (called "safe submission") to begin the semester is to download your files from T-Square after they have been submitted to an empty directory, so that you may recompile and test them. Missing files cannot be given any credit, and non-compiling homework will receive few to zero points. Also recall that late homeworks will not be accepted. Remember that the deadline for the homework is at 11:55 pm, so your program must be submitted before that time or it will be late. Do not wait until the last minute!

Submitted Attachments



[P1Errors.txt](#) (1 KB; Aug 24, 2009 9:39 am)



[GTBuzz.java](#) (1 KB; Aug 24, 2009 9:39 am)