

CS1322 Homework 2

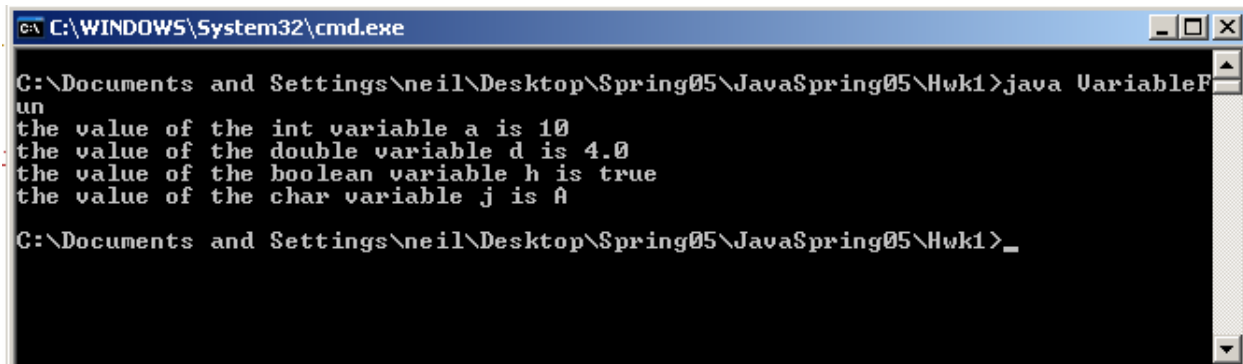
Due: Sunday, January 23, 2005

Introduction

In this homework, you will investigate how to use variables and when to cast them. You will also use a variety of mathematical operators.

2.1: Variables

In lecture, you learned what a variable is and the different types of variables. In this assignment, you will declare variables and print them out. Create a class, similar to the one in assignment 1.1, but call this one `VariableFun` (so your file should be called `VariableFun.java`). In the main method, declare four variables: one of type `int`, one of type `double`, one of type `boolean`, and one of type `char`. Initialize them to an appropriate value. Then print to the console the value of each variable preceded by a descriptive label. Sample output is shown below.



```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\neil\Desktop\Spring05\JavaSpring05\Hwk1>java VariableFun
the value of the int variable a is 10
the value of the double variable d is 4.0
the value of the boolean variable h is true
the value of the char variable j is A
C:\Documents and Settings\neil\Desktop\Spring05\JavaSpring05\Hwk1>_
```

Assignment 2.2: Using variables to implement mathematical formulas

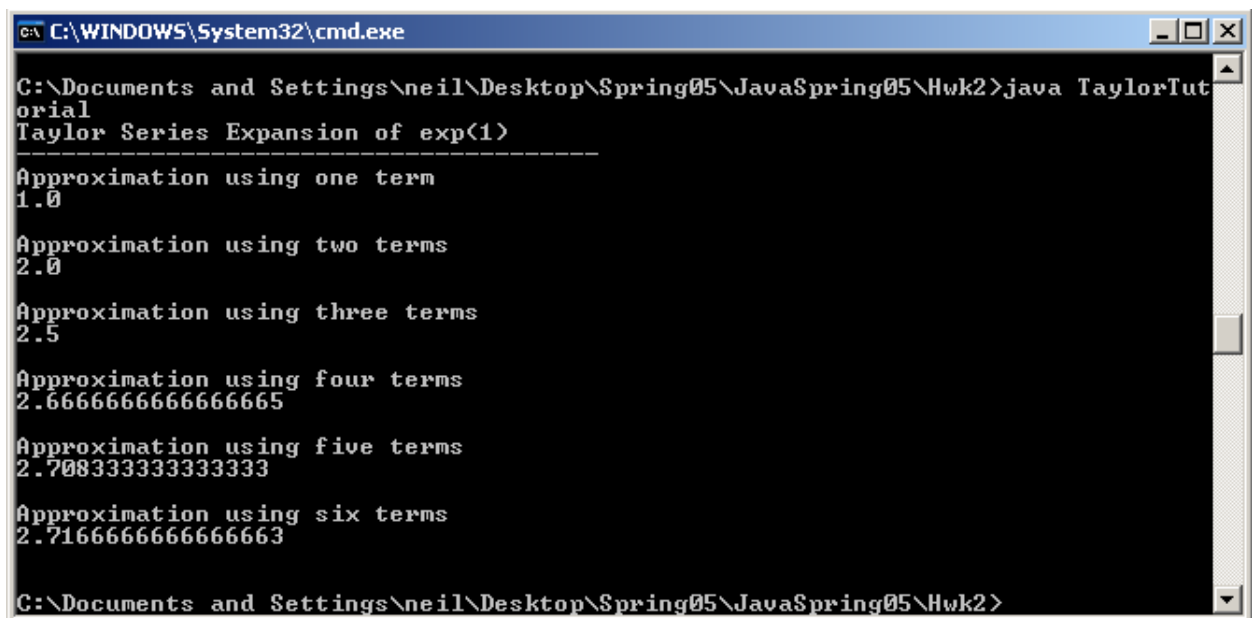
One of the greatest uses of a computer is to compute complex mathematical formulas. In this assignment, you will be modifying a simple method that takes in the number of quality points and the total number of hours taken and calculates the grade point average (GPA). Currently, you have not learned how methods work; nevertheless you will code your GPA calculator inside a method called `computeGPA`. The framework for this method and the class it is in are provided on Webwork under the filename `MathFunctions.java`. The code is also given on the last page of this homework in Appendix A. All you must do is code the one line formula inside the method. The test code has already been written in the main method, which means when you run `MathFunctions`, the appropriate GPA will be printed out. It is imperative that you test your formula with values other than the ones provided in order to ensure that your code is correct. To do this, edit the values assigned to the integer variables 'qpoints' and 'tothours' provided in the main method.

Assignment 2.3: More fun with math

Suppose that you have been asked by your Calculus professor to write a Java program that computes the Taylor series expansion of the function $\exp(x)$ at different degrees of approximation. A Taylor series is an infinite series that when summed up gives the value of the function it represents. Below is the infinite series for the function $\exp(x)$.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \text{for all } x$$

Of course, it would take a very, very long time to sum up infinite terms; therefore, it is best to approximate a function by summing up only a few terms. For this homework, you will do all the math necessary for and print out the approximated value of $\exp(x)$ after summing 1 term, 2 terms, 3 terms and so on all the way to 6 terms. If you are doing this correctly, you should see each approximation become more and more accurate. Below is sample output for $x = 1$. The correct answer of $\exp(1)$ is about 2.7183. Notice how after each stage, the approximation approaches 2.7183. (Remember that $n=0$ is the first term, $n=1$ is the second term, and so on.)



```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\neil\Desktop\Spring05\JavaSpring05\Hwk2>java TaylorTutorial
Taylor Series Expansion of exp(1)
-----
Approximation using one term
1.0
Approximation using two terms
2.0
Approximation using three terms
2.5
Approximation using four terms
2.6666666666666665
Approximation using five terms
2.7083333333333333
Approximation using six terms
2.7166666666666663
C:\Documents and Settings\neil\Desktop\Spring05\JavaSpring05\Hwk2>
```

Since we have not covered the specifics of writing a method, you may code this assignment in the main method. However in the future, writing code other than test code in the main will be unacceptable. We haven't covered exponents or factorials yet, so do those manually in the code you write, e.g., $4! = 1 * 2 * 3 * 4$. Name your file TaylorTutorial.java.

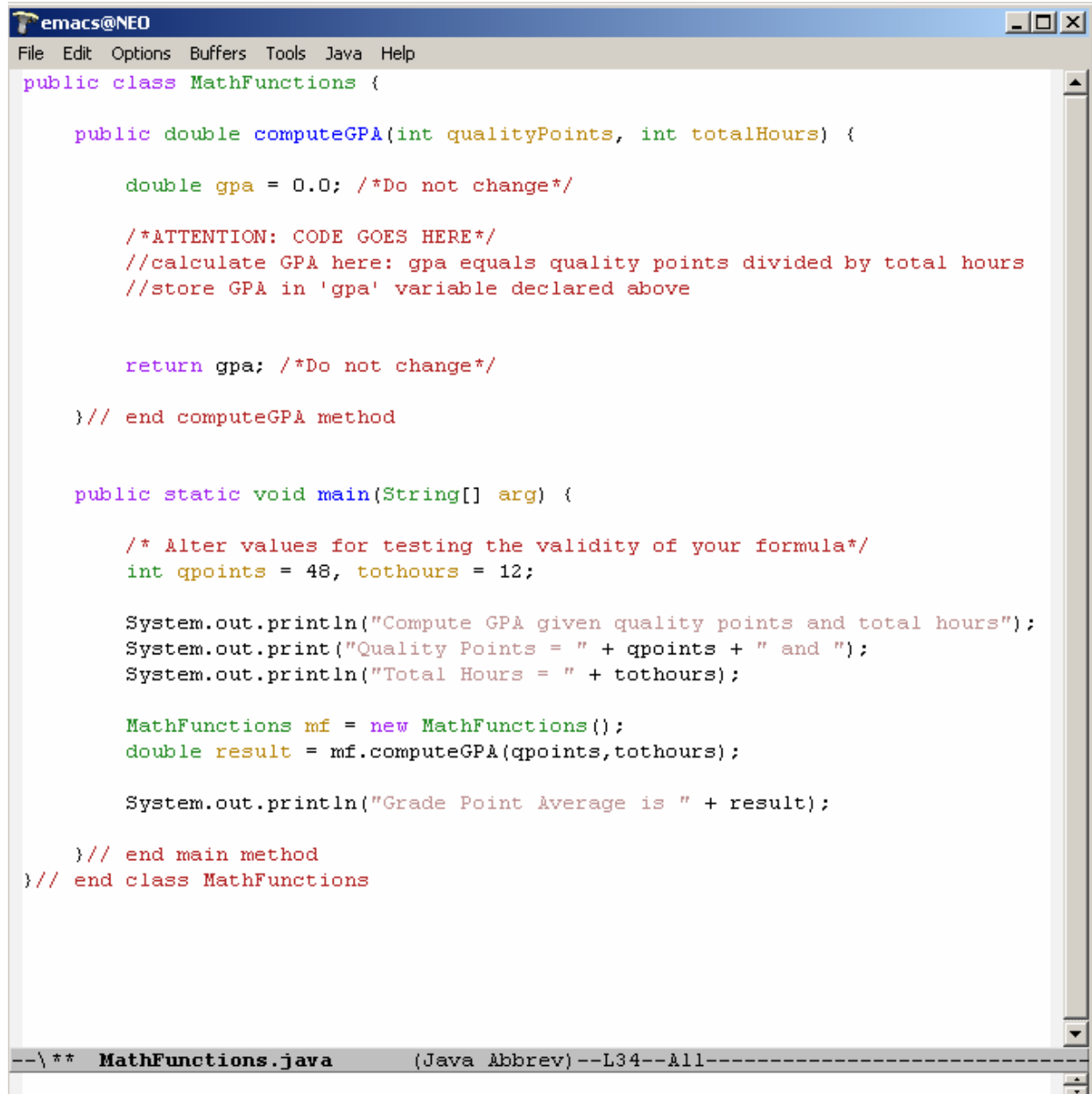
Turn-in Procedures

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

- VariableFun.java
- MathFunctions.java (with modified computeGPA method)
- TaylorTutorial.java

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.

Appendix A: MathFunctions.java



```
emacs@NEO
File Edit Options Buffers Tools Java Help
public class MathFunctions {

    public double computeGPA(int qualityPoints, int totalHours) {

        double gpa = 0.0; /*Do not change*/

        /*ATTENTION: CODE GOES HERE*/
        //calculate GPA here: gpa equals quality points divided by total hours
        //store GPA in 'gpa' variable declared above

        return gpa; /*Do not change*/

    } // end computeGPA method

    public static void main(String[] arg) {

        /* Alter values for testing the validity of your formula*/
        int qpoints = 48, tothours = 12;

        System.out.println("Compute GPA given quality points and total hours");
        System.out.print("Quality Points = " + qpoints + " and ");
        System.out.println("Total Hours = " + tothours);

        MathFunctions mf = new MathFunctions();
        double result = mf.computeGPA(qpoints,tothours);

        System.out.println("Grade Point Average is " + result);

    } // end main method
} // end class MathFunctions

--\ **  MathFunctions.java      (Java Abbrev)--L34--A11-----
```