Laboratory 2

CS 1323, Spring 2015

# Objectives

1. Put your name in a comment in your code. (5 points)
2. Create a project and class in eclipse. (5 points)
3. Create a main program. (10 points)
4. Compile a computer program. (10 points)
5. Execute a computer program. (10 points)
6. Declare integer identifier(s). (10 points)
7. Declare floating point identifier(s). (10 points)
8. Use a constant. (10 points)
9. Perform an arithmetic operation. (10 points)
10. Write output to the console. (10 points)

10 points will be awarded for the documentation of your program. That means using good names for variables, proper and consistent indentation of code, and meaningful use of whitespace.

Section 10: When your program is completed and running, have the teaching assistants check it to get credit for the lab. Both partners should have a copy of the code developed at the end of the laboratory. Also both students should upload their code to Janux. Make sure you put both of your names in your code. If you do not complete your work during lab, the laboratory assignment must be completed and submitted individually on Janux by 11:59 p.m. on January 28.

Section 1: The laboratory assignment must be completed and submitted on Janux by 11:59 p.m. on January 28.

# Description

Write a program to calculate the purchase cost of three different books that are being ordered. You may select your favorite books (prices are available on amazon.com), but I’ve included some of my favorite Java books below. This is how your program could work. You can make artistic changes if you wish, as long as the core computation is the same.

*Purchase 1 copy of Design Patterns: Elements of Reusable Object-Oriented Software by Gamma, Helm, Johnson and Vlissides. Each copy costs $32.46.*

*Purchase 2 copies of Effective Java by Bloch that you would like to purchase. Each copy costs $35.48.*

*Purchase 4 copies of Java Puzzlers: Traps, Pitfalls and Corner Cases by Bloch and Gafter. Each copy costs $27.86.*

*The total cost of your purchases is: $214.86.*

*Sales tax is: $13.97*

*The total cost of your order will be: $228.83*

You should have variables that hold the number of copies of each book purchased, and constants that hold the costs of the books. There should be a variable for the final purchase price. The sales tax rate is 6.5%.

Getting double values to print out nicely like that isn’t easy, and it isn’t required for this project. So if you have lots of extra ugly decimal places hanging around it is just fine. We’ll learn how to make it pretty later. [[1]](#footnote-1)

# Programming Strategy

The first thing to do is to create a project in your workspace. You could call it Laboratory 1.

Then create a class. You could call it ECommerce (no spaces are allowed and it should start with a capital letter). Tell it to include a main program by checking the box.

Eclipse will create a shell for a program that will look something like this:

public class Ecommerce

{

public static void main(String[] args)

{

// Write your code here

}

}

Your code is put where the comment is. Don’t worry about what the stuff above means. We’ll explain it slowly, as the semester goes on.

It is best to write programs methodically. One approach for this program would be to get everything working perfectly for one book, and then copy/paste and edit the code to get the other two books to work.

To show output with variables:

int variable = 5;

System.out.println(“The value of variable is “ + variable);

Also, be sure that you meet all of the learning objectives for the project, since this is how we grade projects.

# Section 10: Paired Programming

We will do paired programming this semester. This is a special way of programming that has been found to help new programmers become better programmers more quickly and with less frustration.

Both partners are working on a single program on a single computer.

**Driver:** Controls the keyboard and the mouse. Enters the program as they think it should be done.

**Navigator:** The navigator watches the driver write the program helps the driver select the best direction to go, and asks questions about what is being done and why. The navigator suggests improvements that could be made to the program. The navigator supports the driver by looking up things in the textbook, or the API (when we get to the point in the semester where we are using the API more). The navigator doesn’t touch the mouse or keyboard where the program is being developed.

The roles of driver and navigator are switched every 20 minutes. Set a timer on your computer (I like Focus Booster from <http://www.focusboosterapp.com/download> but there are lots of timers available).

Caution: It turns out that it is REALLY annoying to have someone jump on every little typo you make before you’ve had a chance to see it and fix it. It’s best for the navigator to wait until the driver has finished a line (including the semicolon) and had a few seconds to read it and fix things that are obviously wrong before pointing out an error.

# Section 10: At the End of Lab

Make sure that both partners have their own individual copy of the lab. Export the Java files in eclipse from the driver--the same thing you have to do to submit them on Janux. The navigator should then import the exported file into eclipse. Get this done at the end of the laboratory so the TAs can help you with the details.

If you don't finish this assignment during the lab time:

1. Have the TAs create a grading sheet so you can submit your code later. Students who do not attend lab cannot submit laboratory assignments.
2. Each person should work on the project on their own after the laboratory is over. We've found that it is very difficult for students to find time to work together after the lab. We also found that some student pairs would have one person finish the lab and submit for both, sometimes even alternating laboratory assignments. This is clearly academic misconduct (both submitting someone else's work as yours and submitting your work as someone else's). So we now dissolve the partnership at the end of lab.
3. Submit your Java code on Janux before the deadline.

1. For students who are bothered by the ugly decimal points. There are two ways to solve this problem. You can use mathematical operations to turn the double price into integer numbers and print them out as integers. This would be done for both the dollars and cents part. Another possibility for people who have done some object oriented programming is to use a DecimalFormat object. Neither of these techniques is required on this project and no extra credit will be given for solving this problem. [↑](#footnote-ref-1)