

Introduction to Computer Science in Java

David Dobervich - Rm. 127
<http://205.173.44.89/>
<http://java.projectwombat.com/>

email: david_dobervich@fuhsd.org

Overview

Unlike other computer courses, this is not an "applications" course where you learn to use a program like photoshop. Computer programming teaches you how to create you own programs for other people to use. It also teaches you how to use the computer to solve a wide array of different kinds of problems in science, business, and everyday life. Finally, it's loads of fun and feels very creative and satisfying to create your own ideas. Programming will give you the ability to do things that will amaze non-programmers.

Learning to program is largely about two different kinds of skill:

Problem solving: If you want to make something or have the computer solve a problem for you, you need to be able to imagine a step-by-step process that would solve your problem or create your idea. For complicated problems, you may need many interacting processes. Computer science teaches you ways of thinking that help you **manage complexity** by understanding a large complicated system in terms of simpler parts.

Learning a language: Second, you need to be able to describe these step-by-step instructions to the computer, so it can do what you want. To do this, you need to learn a language. There are lots of different programming languages (c, c++, python, ruby, lisp, scala, x86 assembly), but we will be using a commonly used language called Java. Once you have learned your first programming language it is much easier to learn more. (I have written programs in 9 languages, for example, and that's not uncommon for experienced programmers).

No previous programming experience is required to take or do well in this class! While programming doesn't feel like doing math, I do require that you have a B or higher in Geometry, Algebra 2, or higher. The reason is that programming requires abstract problem solving, so while it doesn't feel like math, if you can do well in math you can probably do well programming.

I believe the best way to learn to program is to do a lot of programming, and to work with other programs to compare and contrast your approaches as a way to improve. The overall organization of the class typically involves short presentations from me about a new topic, a set of small programming exercises to get a feel for the new topic, and then one or more larger programming assignments that let you apply the idea in a more complex (and interesting!) setting.

You will make many of your own programs from scratch. You will also need to extend and use large existing code frameworks which I will give to you (since this is the norm for most programming now). You will also sometimes pair-program or revise someone else's programs, because understanding how someone else thinks about a problem can expand your own understand and ideas and make you a better programmer.

Specific Course Objectives and Topics

I have one overall course objective: When students leave this course, I want them to have the knowledge, confidence, and independence to 1) learn more programming on their own, and 2) use their programming experience to tackle new problems in their own lives.

Some of the specific topics we will cover are as follows:

- Understand (at a high level) the basic hardware and software architecture of a computer.
- Learn the basics of Java syntax. (variables and assignment, arithmetic operators, conditionals and loops, writing methods, creating classes).
- Learn to use common data structures (arrays v.s. ArrayLists, two-dimensional arrays, HashMaps).
- Learn and use key Object Oriented concepts such as classes and objects, inheritance, abstract classes, and interfaces.
- Learn good programming style.
- Practice applying these skills to a wide variety of different problems.

Materials to Bring to Class

Binder. We will not be following a textbook, so it is very important that students keep and bring handouts to class every day! They should have either a separate 3-ring binder, or a section in a binder which they bring every day.

Flash drive. Students should keep their code on a usb flash drive. 1 gb is more than enough space. I will have flash drives which students can purchase for \$10. If this is a financial hardship, please talk to me privately.

Pen or pencil and paper.

At Home

While many homework assignments can be completed without a computer, it will be of great use for students to have a computer they can program on at home. You can program Java on all major platforms and operating systems, and the software we use is freely available.

Software

Free Software to Install at Home	Location	Instructions
JDK 7	http://www.oracle.com/technetwork/java/javase/downloads/index.html	Mac: Macs come with a JDK pre-installed. So you don't need to do anything! Pc: Download JDK 7 (the left-hand link) for your platform and install!
Eclipse	http://www.eclipse.org/downloads/packages/eclipse-classic-37/indigor	Select the appropriate download. There is no "installer". Just unzip the file in your root folder. The application (eclipse.exe) is in the root directory.
Processing	http://processing.org/download/	Select the appropriate download. (Do not download the version without Java). As with Eclipse, unzip it in your root folder.
If you need more help installing these programs, please email me! I'm happy to help.		

Grading

Basic knowledge of the java language (quizzes)	30%
Support work (homework/classwork)	20%
Programming ability (projects)	30%
Midterm / Final Project	10% each

Basic knowledge of the java language just means whether or not you know the basic concepts and constructions. For example, do you know what all the parts of a for-loop are. I will assess this mainly with short quizzes and tests that will involve short answers and very very short programming exercises.

You are always welcome to re-write and re-submit your code, with the caveat that you can only re-submit one item per week. (In other words, you may not re-submit all of your work in the last week before a grading period).

How to improve your grade

I am very happy to give you opportunities to raise your grade! UNLESS you only sort-of do the home-works, and your participation in class is kind of passive and you ask me at the last minute.

I will not discuss your grade during the week before the grading period ends. I will be very happy to talk about your grade at any other time! I cannot give you a checklist of things to guarantee a certain grade if you complete them. I can give you opportunities to demonstrate to me that you understand the ideas I'm interested in; and if you know Java at a certain grade level, I will be happy to adjust your grade.

Lab Policies

No food or beverages except bottled water.

Do not plug or unplug anything without asking me first. Even if you are trying to fix something, ASK ME FIRST.

Do not alter anything about the software on the computer. This includes things like the desktop.

You must be programming while on the computers. It's ok to use the web to reference documentation, but other non-programming activities will result in you having to complete programming assignments without the computer, and an escalating series of consequences. The good news is that programming is fun, so this shouldn't be a problem.

A Note About Collaborating

I encourage you to collaborate with each other. This is, in fact, different from copying. Here is a rough guide to help.

Because copying can result in very serious consequences, if in doubt, just ask me if you have a question about whether it's "ok" to do something and I'll let you know.

Collaborating -- encouraged!	Copying -- very serious
Talking about ideas. Sketching out code on paper or the whiteboard. Letting someone point out some fixes to the code on your computer and YOU do the actual changes.	Copy-and-pasting other people's code, even if you intend to modify it. (note: I may sometimes instruct you to use someone else's code for a specific purpose; if I've said to, then it's ok). Letting someone else type something at your keyboard.

Readings

While we will not be following a specific textbook, there are lots of excellent resources available which students are encouraged to look at.

- An excellent free online textbook called *Think Java* can be found at <http://greenteapress.com/thinkajava/>
- Sun Microsystems has a good set of Java tutorials which can be found at: <http://download.oracle.com/javase/tutorial/reallybigindex.html>
- If you wish to have a book to reference on your own, I recommend Java: How to Program by Deitel & Deitel as a comprehensive and clear guide.

Resources & Other Things

An excellent website to search if you have a programming question is StackOverflow: stackoverflow.com

We will be using the *Processing* environment a lot in this course. It's a Java framework to facilitate the rapid development of simple graphical interfaces and data visualizations. It is extremely awesome. Their site is very well organized with lots of examples and tutorials. It can be found at: processing.org