CS 112 – MiraCosta College

Introduction to Computer Science II
Java

Module 1 – Graphics

Chris Merrill
Computer Science Dept.
cmerrill@miracosta.edu

Agenda

- My introduction
- What you’ll need and should already know
- Syllabus, Assignment List, Canvas
- Attendance
- Course Outline
- CS 111 Review Quiz
- Java’s Swing (Graphics) Package
- Group Lab 1 – Graphics

“Critical” Thinking

“We live in a society that increasingly makes it easy for people to get through the day without having to think very much. We have microwaveable food, entertainment at our fingertips, and GPS to get us where we need to go. I’m not saying those things are bad. Ideally, such time-saving devices free up our brains for other, more important pursuits. But the practical effect is that we’ve become accustomed to setting our brains on autopilot…”

"…Actual thinking requires deep and protracted exposure to the subject matter — through close reading, for example, or observation. It entails collecting, examining, and evaluating evidence, and then questioning assumptions, making connections, formulating hypotheses, and testing them. It culminates in clear, concise, detailed, and well-reasoned arguments that go beyond theory to practical application."

Rob Jenkins
Professor at Georgia State University

“Computational” Thinking

One of the exciting things about learning Computer Science is that you learn a new and fundamental way of thinking and problem solving; a way of thinking that is critical in the 21st Century. It is called "Computational Thinking" and the idea that this is one of the big advantages of studying Computer Science, whatever your ultimate career, is causing a big stir..

…Computational Thinking is a collection of diverse skills to do with problem solving that result from studying the nature of computation. It includes some obviously important skills that most subjects help develop, like *critical thinking, creativity, the ability to explain*, and *teamwork*. It also consists of some very specific problem solving skills such as the ability to think *logically*, *algorithmically*, and *recursively*. It is also about understanding people. Computer Science is unique in the way it brings all these diverse skills together.

“Computer Science For Fun”

Your Instructor

- Chris Merrill
  - A.B. Dartmouth, M.S.E. University of Michigan
  - Developed apps for 35+ years (managing for 25)
    - typically working on large business systems
- For this course
  - Email me at *cmerrill@miracosta.edu*
  - Office hours: Thursday 10:30 – 11, Friday 12:30 – 1pm
  - Your first line of defense is the Computer Lab in the Hub
  - *Strongly* suggest using MiraCosta's tutoring services

What You’ll Need

- An email address
- A “Print” account at MiraCosta (preferable)
- Access to a computer outside of this classroom
    - either at home or to those available in the Hub’s Computer Lab
    - with the ability to print your homework and lab assignments
    - with access to the Internet

Textbook

Absolute Java, 5th *or* 6th edition, by Walter Savitch

- Either edition is fine, as projects and contents generally don’t change between the two editions
- Copies are available in Hub (Library)
- I use several of the slides provided by Pearson which summarize covered chapters

What You Should Already Know

- Primitive data types
- Variables and constants
- Branching and looping statements and constructs
- Classes, constructors, and methods
- Arrays (creation, passing to methods)
- Inheritance (sub and super classes)
- Encapsulation (**`public`**, **`private`**, **`protected`**)

Quiz to follow later today…

Software

- IDE (Integrated Development Environment)
  - JCreator (preferred) – www.jcreator.com (Windows only)
  - Geany – (Win or Macs)
  - Please do NOT use Eclipse unless you already use it
- If using your own computer, please download and install an IDE by next class
  - Use instructions on Canvas
- Ditto for Java – www.java.com
  - Need to download JVM (Java Virtual Machine) on your home machine

Virtual Machine

- Alternately, you can use a “virtual machine” (VM) provided by MCC, where you are actually using a computer in the “cloud” to create, store, and run your programs.
- Your virtual machine can be accessed from any computer connected to the internet.
- I’ll be using my virtual machine during most of the lectures for this class.
- We’ll review this later today.

Syllabus and Assignment Handouts

- Course Goals and Objectives (SLO's) from Syllabus
- Grading
  - Most labs (projects) will be performed in groups
  - Homework and tests must be done *individually*
- Important dates
  - I'll let you know one Module in advance of a quiz
  - Midterm April 6
  - Final exams and final project demo May 25
- No makeup quizzes or exams unless previously arranged!
- Review Readings & Assignments handout

Workload Expectations

- Every class will have a 2 to 3-hour lecture
- There will be homework every Module except Midterm
  - It should take 4 to 5 hours to complete
- Time will be provided in class for Group Labs
  - You may have to meet outside of class to complete
- Study the appropriate textbook sections
- Don’t wait until the last minute to ask questions.
- Extra Credit opportunities will be provided

Attendance

- Roll Call
  - Normally 30-60 minutes into the class
- Adds
  - You must be enrolled by the end of the 2nd week
- No crashers, please…

First Announcement on Canvas

- Enrollment
- Drops
- General expectations

Canvas

- Find course presentations, handouts, scripts, and canned Java classes on Canvas

- Homework is due at the ***beginning*** of class following the day the assignment is made

- I'll usually have homework grades posted within 3-4 days of the due date.

Course Modules – 1st half

- Review of topics from CS 111 – today
- Introduction to Graphics – today
- Polymorphism / Dynamic Binding, Abstract Methods
- Copy Constructors, the clone method
- Exception Handling
- I/O Streams, Binary Files
- Recursion
- Sorting
- UML

Course Modules – 2nd half

- Interfaces
- Inner classes
- Generics
- Dynamic data structures and linked lists
- Iterators
- Graphic User Interfaces (GUI)
  - The Model / View / Controller (MVC) Pattern
- Containers, JPanel
- Event-driven programming
- Fonts, icons, and scroll bars
- Threads

My "30-minute Rule"

- Don't spend more than *30 minutes* on any one issue or problem. My intention is *not* to make you struggle hopelessly through homework problems, exercises, labs, or projects
- If you can't resolve a specific issue or problem after 30 minutes, take a break…
- After another 30 minutes, if you *still* can't solve the problem, please email me, and include all applicable work as an attachment.

Do Not Get Ahead!

I know that some of you have programmed before in Java outside of CS 111 and CS 112.

If you want to add functionality to homework, your lab, or your final project, that’s great

BUT

Please do *not* use techniques or features of the Java language that we haven’t yet discussed in class.

Do Not Get Ahead!

Please do *not* use the following features of Java until we cover them in class

- ArrayLists
- Linked Lists
- Containers
- Iterators
- Collections
- Recursion
- Abstract Classes
- Interfaces
- Adapters
- Inner Classes
- Generics
- Threads

Today’s Quiz

- This quiz tests what you already know about Java
- When done, we’ll go over the answers
  - I won’t be recording grades
- Please review those questions that you missed before next class

Java Class Requirements

A few requirements that I have for Java classes

(We’ll discuss “style sheets” next class.)

Java Application Programs

- Two common types of Java programs are *applications* and *applets*
- A Java *application program* or "regular" Java program is a class with a method named **`main`**
  - When a Java application program is run, the *run-time system* automatically invokes the method named **`main`**
  - All Java application programs start with the **`main`** method

Applets

A Java *applet* (*little Java application*) is a Java program that is meant to be run from a Web browser:

- it does *not* have a `main` method
- it uses a windowing interface
- it typically is run from a location on the Internet, but can be run with an applet viewer for debugging purposes
- it is invoked via HTML program

Unfortunately, applets are on their way out

- No support from Chrome or Firefox

Swing

In contrast, application programs may use either a windowing interface *or* console (text) I/O

- In today's lab, we'll use a graphics package named Swing
- It requires a **`main`** method, but no HTML code
- It too is being replaced by a new package named JavaFX
  - JavaFX extends Swing to replace applet capabilities and to make it more internet-savy.

Let’s Get Going with Graphics

- Configuring JCreator
- Running your first Java graphics program
- Canvas pages
  - Creating a graphics program in JCreator

Homework

- Email me a message letting me know that you intend to take this class
- Acquire a textbook
- Ensure that you can log onto Canvas for this class
- Set up a print account with $5 in it at the Library Hub
- Load Virtual Machine software on your system at home
- If you want to run locally, then load JCreator, Geany, or BlueJ, and Java on your system at home

… *AND* …

Homework

- Review today’s quiz
- Review the Graphics Supplement Handout, and complete any unfinished work from Lab 1 with your group.
- Complete Homework for Module 1 (Java Graphics)
- Read chapter 8 – Polymorphism and Abstract Classes
- Quiz 2 in two weeks on Polymorphism

Group Lab

Follow the lab in Canvas. You'll be using Java's Swing package to draw several images.

Create a single Word document containing PrintScreens of the results or running your programs (no Java code needed), plus the answers to 2 questions found in the Lab.

Place the names of all group members at the top of your Word document and turn in next week.

Finish the parts of the lab not completed by your group individually and submit with your homework next week.