Introduction to Programming
Why Bother Programming?

- Every application on your computer is a program!

- We’ll learn the basics of languages.
  - A language is used to write a program/application.
  - Later classes cover more advanced features.

- Your programs will be the same as Microsoft Word, Excel, Photoshop, Grand Theft Auto, etc.
  - You write code (the program).
  - Click on it (or type in a command).
  - See results on screen.
My Programs Look Cheesy 😞

- Your code won’t *look* like “Grand Theft Auto”.
  - maybe “Pong”

- Why?
  - lack of experience
  - lack of time
  - lack of tools
  - not a graphics class

- We gotta’ start with the basics.
  - but you can pick up cool graphics easily
Our Programming Philosophy

In this class
- will learn basics that apply to all languages
- will use Java
- will be aware of other languages

You will find that it is very easy to transport this knowledge to any other language.
Ok, Let’s Write a Program!

Follow what I do on the screen.

- Open Eclipse.
- Create a new project.
- Create a new class.
- Type in your program (see next slide).
Here’s What You Type

Get it *exactly* as shown. Punctuation, caps, everything.

```java
public class HelloProgram
{
    public static void main(String[] args)
    {
        System.out.println("Hi there!");
    }
}
```
Run It

Next to the green arrow is a little black arrow. Press it.

Select “Run as…” and choose “Java Application”

- See the result on the console?

- Ok, I admit. That was boring…
Let’s Improve Our Code

Create a new class called CoolHello.

Type in the program on the following slide.

- Don’t worry that this stuff is mysterious.
- Just want you too see it’s easy to get started.
- All will be explained later…

Then run it, like before.
import javax.swing.JFrame;
import javax.swing.JLabel;

public class CoolHello {
    public static void main(String[] args) {
        JFrame myWindow = new JFrame();
        myWindow.setSize(300, 400);
        myWindow.setTitle("Cool Hello");
        JLabel label = new JLabel("Yo, wassup?");
        myWindow.add(label);
        myWindow.setVisible(true);
    }
}
Don’t Need Eclipse

- It’s just a fancy tool.
  - Contains a WYSIWYG text editor.

- We can also do it all from the command line.
Programs Are Just Files

- Open Notepad.

- Type in the simple HelloProgram.
  - Get all of that punctuation, caps, etc. *exactly* correct!

- Save as HelloProgram.java
  - Do NOT save as HelloProgram.txt
  - Do NOT save as HelloProgram.java.txt

- Save on the C: drive.
Java is a File

- The code was saved in a file called “HelloProgram.java”

- just like “MyHorribleEnglishEssay.doc”
  - Microsoft Word documents end in “.doc”

- or “MyExcelSpreadsheet.xls”
  - ends in “.xls”
Run From the Command Prompt

The Command Prompt program on your computer. You can find it under All Programs → Accessories.

![Command Prompt window](image)
Umm, What’s That “javac”?

- **javac** is the command to **compile** the program.
  - This creates an intermediate file called HelloProgram.class that is ready to be run as a program.

- **java** is the command to run the program.

- Most languages do something similar
  - `cc HelloProgram.c`

- So how’s this compiling business work?
So How’s a Program Work? (part 1)

You write code that looks like

```java
public class HelloProgram {
    public static void main(String[] args) {
        System.out.println("Hi there!");
    }
}
```
A compiler turns it into assembly language.

- Assembly language is created by the CPU manufacturer for your particular type of chip (e.g., by Intel).

- Most languages like C++, Visual Basic, etc. will be converted to assembly language.
  - Java has a wrinkle – stay tuned…

```
public class MyTaxProgram {
    ...
}
```
So How’s a Program Work? (part 3)

The assembly language is converted to machine language.

- This is the set of 0’s and 1’s that are understood by your CPU.
- In reality, computers *only* understand 0’s and 1’s.
  - Your chip uses +5 and -5 volts to represent these numbers.

**Assembler**

```
LOAD   COST
ADD    TAX
STORE  FINALBILL
```

01001100

Computer hardware
But There’s a Major Problem!

- Every type of CPU has a different machine language.

- So if I write a program in C++ on one computer, it won’t run on some other computer.
  - Have to rewrite, recompile, and reassemble your program on every type of computer.
  - Pain in the &^%$$#!.
Java adds an interpreter called the Java Virtual Machine (JVM).

- The compiler converts your code to “byte code”.
  - 0’s and 1’s like machine language, but it’s the same 0’s and 1’s for every computer!
- The JVM then converts that to the machine language for your particular hardware.

Why is this different? The same Java code can now run on any machine.

- All you need is the JVM for your particular computer (or cell phone, game console, toaster, roller coaster, car, etc.!)
See Why .coms Like Java?

- Write once, run everywhere.
  - Same code can run on billions of different devices from tv's to microwaves.
  - Lower costs!
  - Also excellent for the web.
  - Also interacts well with databases, xml, etc.
There’s More.

Java also frequently uses Just In Time Compilers (JITs).

- These don’t compile your code until the moment it is needed.
  - So it doesn’t care if it is on a car or a Martian rover.
  - Compiles at the last minute and then decides what it needs to do to make it work on that particular machine.
Ack! Too Much!

For now, just remember (1) the basic picture and (2) that Java improves with an interpreter.
Interpreter Versus Assembler

- **Interpreter** works “at run time”.
  - works whenever you ask the code to run
  - does the work *every* time
  - so slower!
  - but works anywhere!

- **Assembler** works before code is run.
  - works once
  - never has to work again
  - so faster!
  - but works only on one computer!

“Write once, run anywhere!”

“Write once, rewrite everywhere!”
Java Good News/Bad News

Advantages:
- write once run everywhere
- clean, easy to understand
- object-oriented (more later)

Disadvantages:
- Slower
- Like all languages, will be replaced eventually
Java’s Primary Competition

- C++
  - fast
  - OO
  - good graphics (video game language)
  - not portable
  - Unix compatibility