

CS210
Intermediate Computing with
Data Structures (Java)

Chris Kelly
ckelly@cs.umb.edu
cg.kelly2013@gmail.com

Prerequisites

- I expect that you already know the Python programming language at an introductory level (usually gained by taking our CS110)
- You should be familiar with the following, in some higher-level programming language:
 - Data types
 - Variables
 - Expressions
 - Statements
 - Selection
 - Iteration
 - Functions
 - Other Tools

Prerequisites

- If not, you should NOT take this course
 - See your advisor for other options such as retaking CS110 or taking CS119
- Reading for this lecture: **L&C 1.1 – 1.2**
- Homework #1: Read assignment for next week

Why Java?

- I liked Python in CS110. It's easy to use. Why do I need to learn Java? It's harder!
- In our ABET accreditation, we state that our graduates will know two languages high in industry usage, i.e. Java and C
- The marketability of the programming languages that you know is critical for getting a job!
- But, how to measure that?

Why Java?

- Four IEEE top ranked languages are:
 - Java, C, C++, Python
 - Source: <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>
- Two most popular programming languages based on Google tutorial searches
 - Java (24.1%)
 - Python (12.1%)
 - Source: <http://pyp1.github.io/PYPL.html>

Why Java?

- The TIOBE index graph from 2002 to present indicates a decades-long Java and C competition for the top position, with Objective-C coming on strong recently due to its use in Apple products
- Python is not even close to the top
- Depending on how you look at it, a language may have all the latest buzz, but not be the most dominant language in industry use
- Source: <https://www.tiobe.com/tiobe-index>

Why Java?

- My take on programming languages is that they come and go, but leave a huge legacy
- There are probably trillions of lines of Java and C code in production use today that won't be abandoned or rewritten in Python
- When you look for a job, you will limit your options by concentrating on the latest fads and ignoring these traditional languages

Introduction to Java

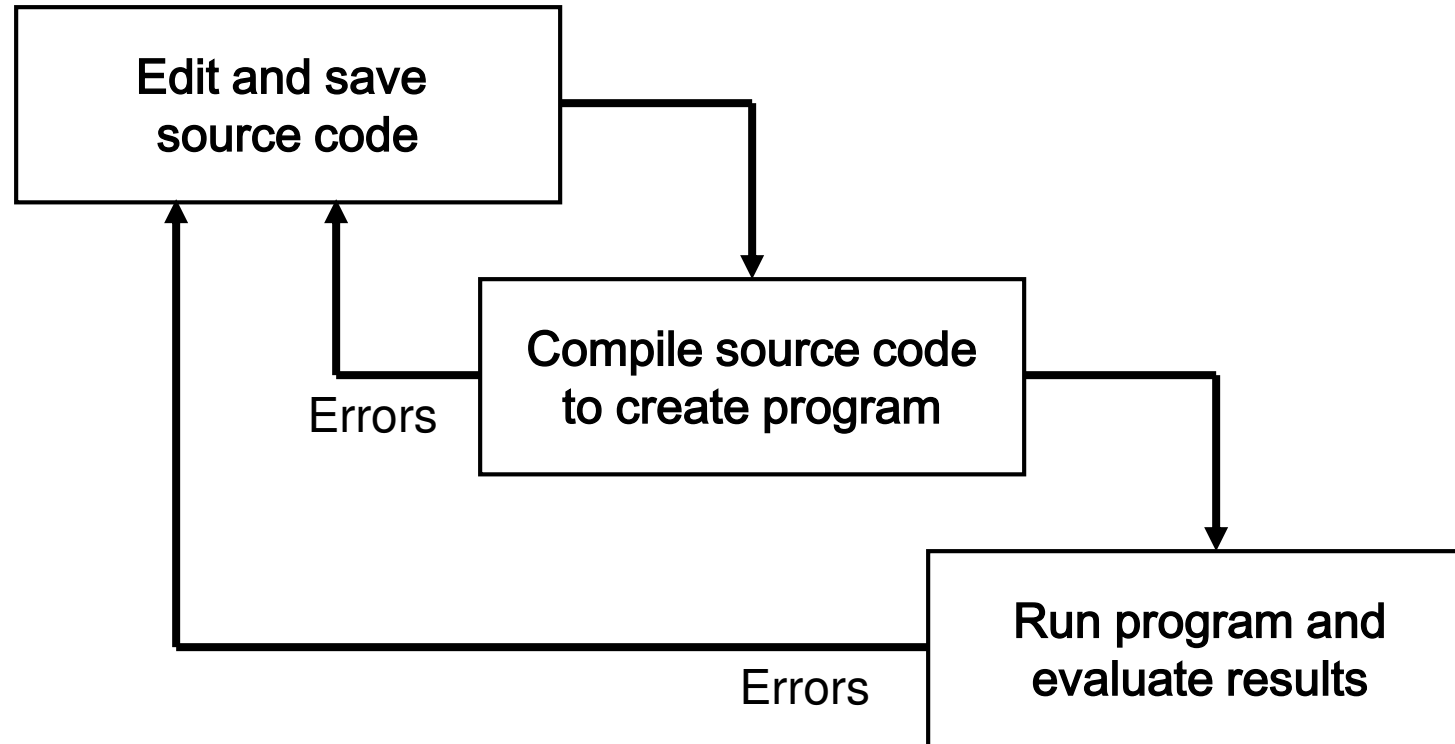
- Programming in Java
- Software Development Tools
 - Oracle/Sun Java Development Kit (JDK)
 - Dr Java
- Hello World example

Programming in Java

- A Java program consists of multiple classes
- A programmer enters each source file (one class per .java file) and compiles all of them
- From each source, the Java compiler produces a “byte code” representation (a .class file) or provides compiler error messages without generating a .class file
- A programmer must correct all compilation errors before executing the program

Program Development Steps

- Classical "Waterfall" Development Steps



Programming in Java

- A Java program can have three types of errors:
 - The compiler will find syntax errors and other basic problems (*compile-time errors*). If so, an executable version of the program (.class file) is not created
 - Problems can occur during program execution, such as trying to divide by zero, which cause the program to terminate abnormally (*run-time errors*)
 - A program may run, but produce incorrect results, perhaps using an incorrect formula (*logical errors*)
- You must learn to fix any level of error occurring

Programming in Java

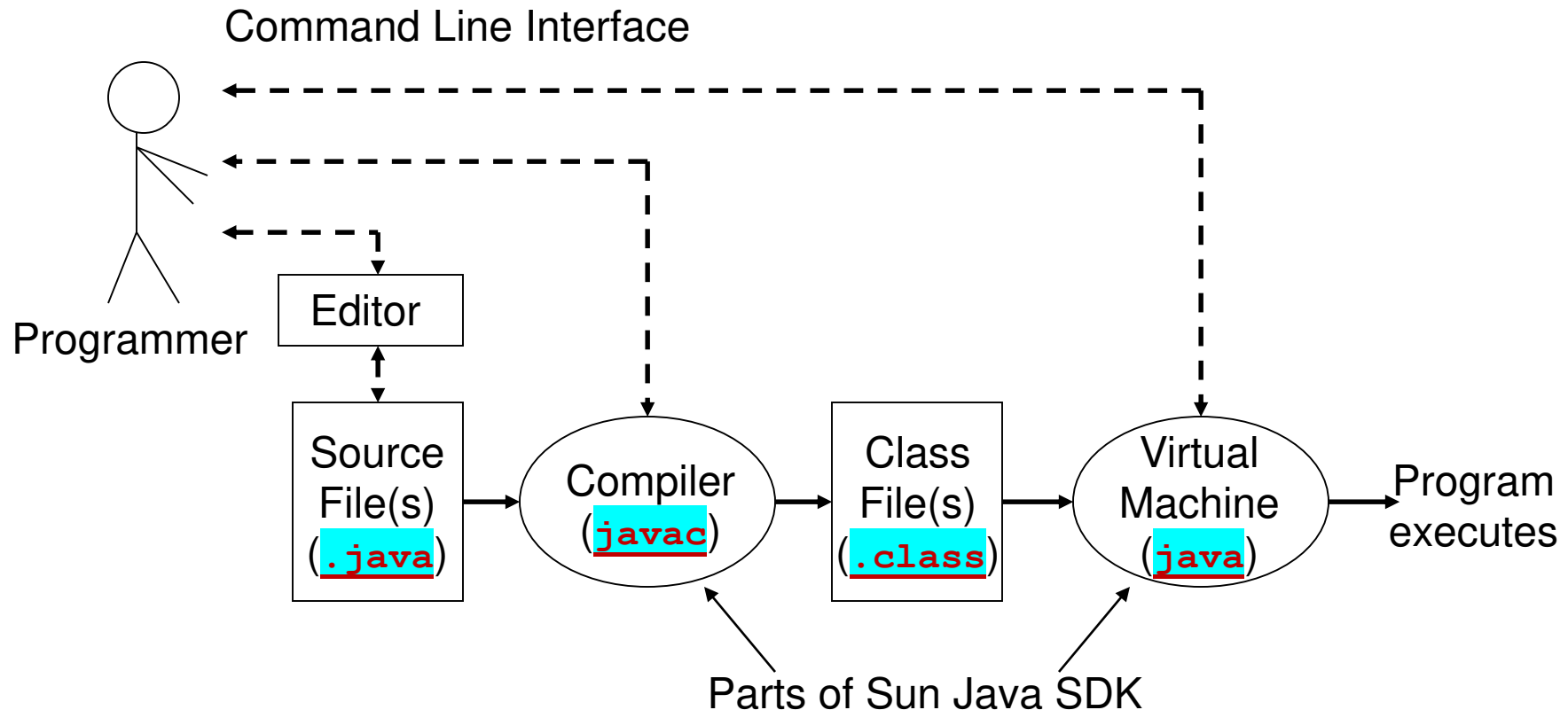
- One of the classes contains a main method:

```
public static void main(String [ ] args)
```

- The Java Virtual Machine (JVM) executes the .class files starting at the main method
- The main method can invoke other methods in the same class and/or instantiate objects of other classes and invoke their methods

Software Development Tools

- Using Sun Java SDK alone



Using Sun Java SDK Alone

- Example DOS Commands and Parameters

```
C:\ > edit HelloWorld.java
```

(Create/edit "source file" in an external window)

```
C:\ > javac HelloWorld.java (creates .class file)
```

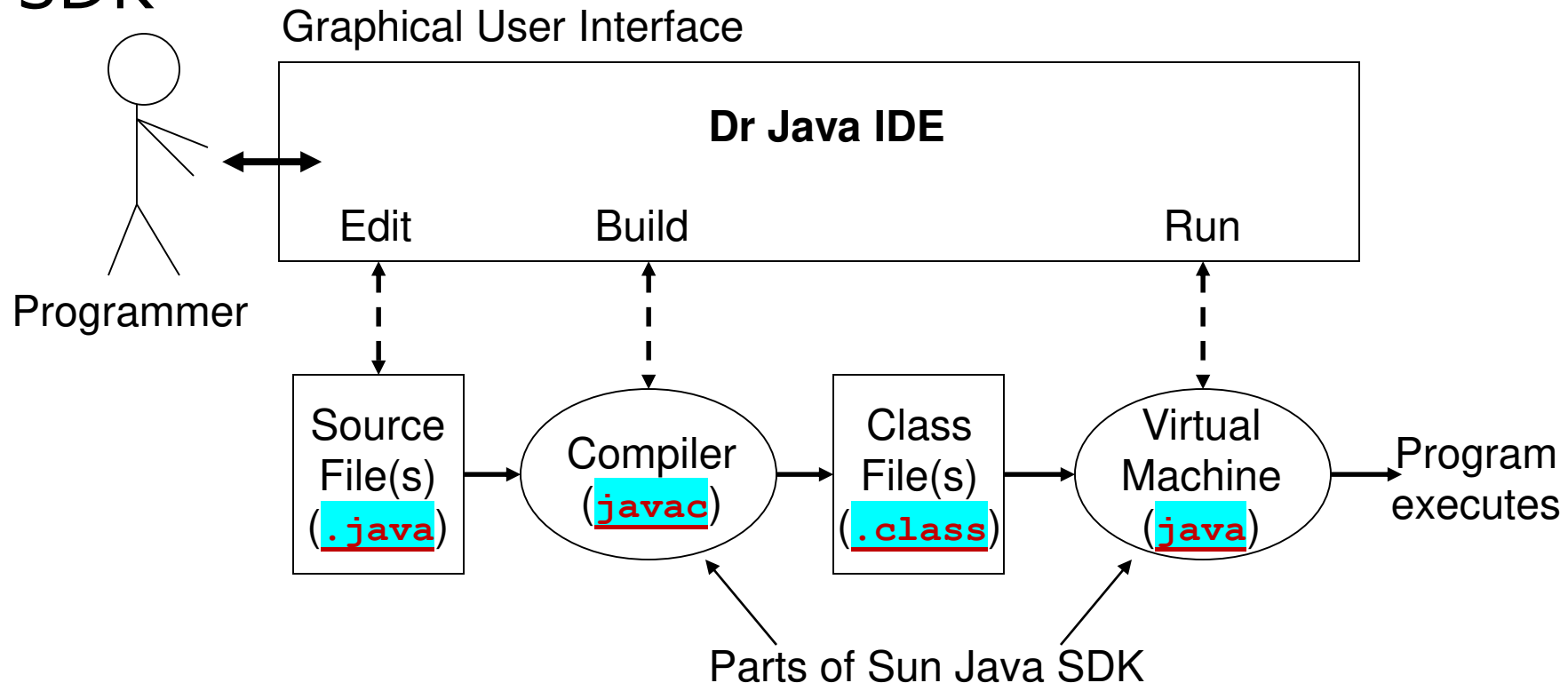
```
C:\ > java -classpath ... HelloWorld
```

```
Hello World
```

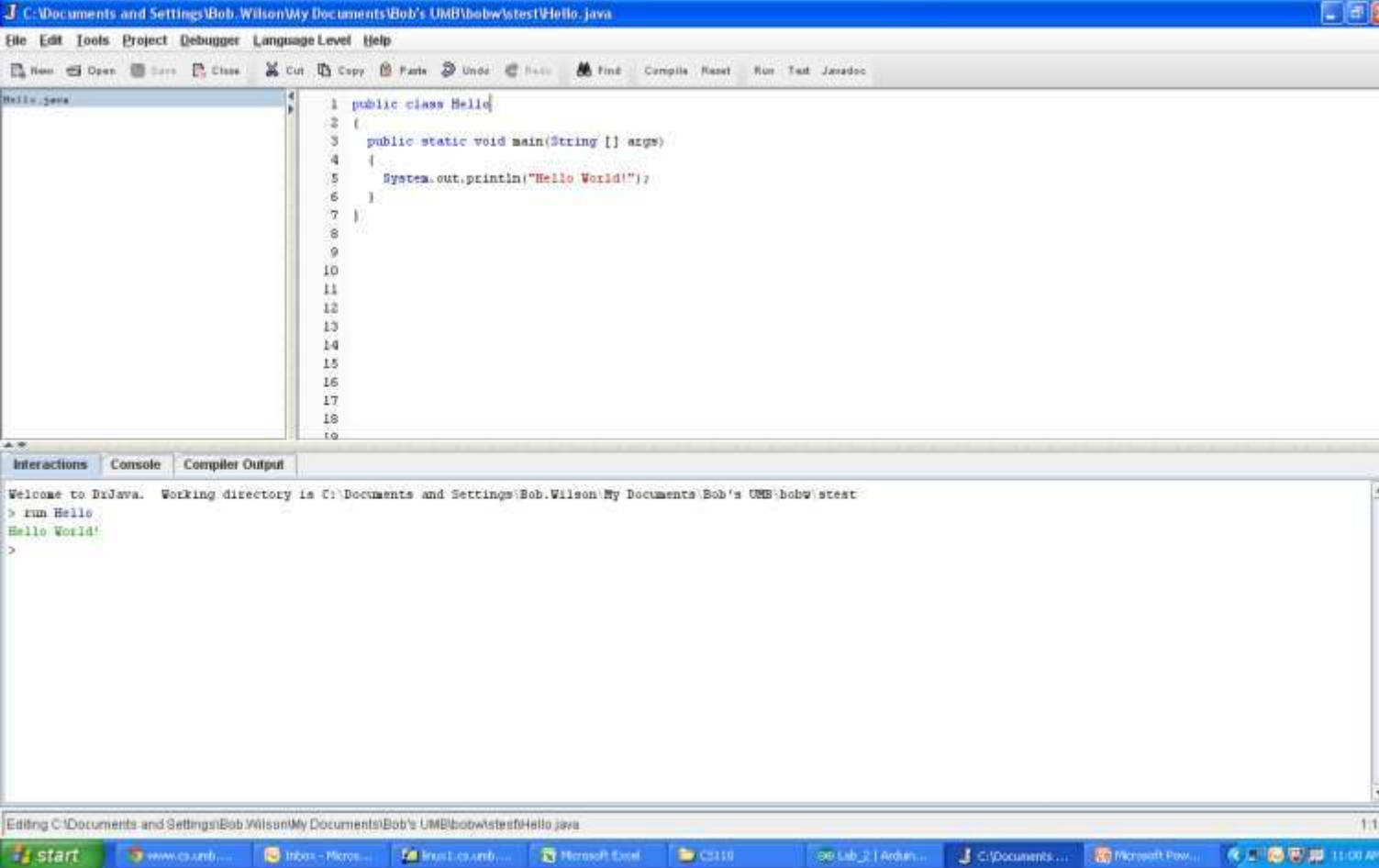
```
C:\ > exit
```

Software Development Tools

- We will use a combination of the Dr Java IDE and the Sun Java SDK



On-line Demonstration: Dr Java



The screenshot displays the DrJava IDE interface. The main editor window shows a Java file named `Hello.java` with the following code:

```
1 public class Hello {
2 {
3     public static void main(String [] args) {
4     {
5         System.out.println("Hello World!");
6     }
7 }
8 }
9
10
11
12
13
14
15
16
17
18
19
```

Below the editor, the `Console` tab is active, showing the following output:

```
Welcome to DrJava. Working directory is C:\Documents and Settings\Bob.Wilson\My Documents\Bob's UMB\bobw\stest
> run Hello
Hello World!
>
```

The status bar at the bottom indicates the current file is `C:\Documents and Settings\Bob.Wilson\My Documents\Bob's UMB\bobw\stest\Hello.java` and the time is 1:18. The Windows taskbar at the very bottom shows the Start button and several open applications including Internet Explorer, Microsoft Excel, and DrJava.

Software Development Tools

- Download/install the software development tools on your own PC if you wish to do your lab and project assignments at home
 - Oracle/Sun Software Development Kit (SDK)
 - Dr Java Integrated Development Environment (IDE)
 - Junit class library (to support testing)
- Use the PCs in the Healey Library labs as they already have these tools installed

Java “Hello World” Program

- Traditionally the first program taught!

```
public class HelloWorld {  
  
    public static void main (String [ ] args) {  
        System.out.println("Hello World");  
    }  
}
```

- Enter, compile, and execute this Java class to test the development environment on your own PC