Why take CS230?

- You will learn the “big picture” of programming
  - Data abstraction
  - Modularity
  - Performance Analysis
  - Basic abstract data types (ADTs)
- You will become a more competent programmer
  - You will also become a designer, tester, analyzer, debugger
  - Also, a good team member
- You will develop a project worth showing off
- You will have fun in the process!

Why use ADTs?

- Allows you to write complex programs more easily
  - To keep mental track of complex data interaction
  - To reuse code
  - To improve code performance
- Allows modularity of large projects
  - Easier to understand large chunks of code
  - Easier to collaborate with large teams
- Basic ADTs
  - Collection
  - Linked List
  - Stack
  - Queue
- Advanced ADTs:
  - Tree
  - Priority Queue
  - Set
  - Map
  - Graph

How you learn and succeed

- Expect to spend at least 15 hours a week
- Read the book weekly.
  - Online copy, 10 paper copies.
- Go to Labs to understand what we discuss in lecture
- Go to SI sessions to solidify your understanding.
- Do the weekly quizzes to verify your understanding
- Take the exams to evaluate your skill
  - Two midterm exams, one final exam
- Do the project to proudly show off your skills!
Collaboration Policy

Registered? You are a member of the group: CS230-F19
- Post questions, answer them if you know the answer!
- Assignment 1 is available and due Thursday at 11:59pm
  - See schedule for link to assignment description
  - You should have an account on the cs server with subdirectories to upload your assignments:
    http://cs.wellesley.edu/~username
- Reading for next class
  - Java Foundations, by Lewis, DePasquale and Chase, edition >= 2. (Denoted as LDC on the schedule)
  - Read Chapters 1, 2 for information about today’s topics
  - Read Chapters 3 and 4 for next time (3.6, 3.7 optional)

Java has Classes and Objects

- A class is like a blueprint from which you can create many of the "same" house with different characteristics

A First Program: Motto.java

```java
/**
 * Our first CS230 program.
 * It prints out Wellesley's motto.
 * @author Orit Shaer
 */
public class Motto {
    // Program execution begins with the "main" method
    public static void main(String[] args) {
        System.out.println("Non ministrari");
        System.out.println("sed ministrare");
    }
}
```

A Java "method" is similar to a Python "function."

Curly braces, rather than indentation, indicate the body of classes, methods, loops, and conditionals

Multi-line JavaDoc comment

Single line comment

String denoted by double quotes

Statements end with semicolons

System.out.println is similar to Python’s "print" function

A public class must be in a Java file with the same name

A Java "method" is similar to a Python "function."

System.out.println("Non ministrari");

System.out.println("sed ministrare");
Interpreters
Source code
/* Our first CS230 program.
* It prints out Wellesley's motto.
*/
public class Motto {
    // Program execution begins with the "main" method
    public static void main(String[] args) {
        System.out.println("Non ministrari");
        System.out.println("sed ministrare");
    }
}

Compilers
Source code
/* Our first CS230 program.
* It prints out Wellesley's motto.
*/
public class Motto {
    // Program execution begins with the "main" method
    public static void main(String[] args) {
        System.out.println("Non ministrari");
        System.out.println("sed ministrare");
    }
}

Java Does Both
Source code (.java)
/* Our first CS230 program.
* It prints out Wellesley's motto.
*/
public class Motto {
    // Program execution begins with the "main" method
    public static void main(String[] args) {
        System.out.println("Non ministrari");
        System.out.println("sed ministrare");
    }
}

Using Java and BlueJ
- You can study **data structures** using any language, but in this course we will use **Java**
- You can **write and execute Java programs** in many ways, on the command line or with an IDE, but in this course we will use a simple IDE called **BlueJ**
Data Types in Java

- Java is a **statically typed** language
  - You must define the type of each variable when it is declared
  - Unlike Python, not all variables in Java are objects
  - Some are **primitive data types** (but have related objects)

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Storage</th>
<th>Range of values</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>32 bits</td>
<td>-2,147,483,648 to 2,147,483,647</td>
</tr>
<tr>
<td>long</td>
<td>64 bits</td>
<td>9,223,372,036,854,775,808 to 9,223,372,036,854,775,807</td>
</tr>
<tr>
<td>float</td>
<td>32 bits</td>
<td>Approx. –3.4E+38 to +3.4E+38 with 7 significant digits</td>
</tr>
<tr>
<td>double</td>
<td>64 bits</td>
<td>Approx. –1.7E+308 to +1.7E308 with 15 significant digits</td>
</tr>
<tr>
<td>char</td>
<td>16 bits</td>
<td>65,535 Unicode characters</td>
</tr>
<tr>
<td>boolean</td>
<td>1 bit</td>
<td>true or false</td>
</tr>
</tbody>
</table>

Variable Declaration in Java

- A **variable** must be declared once before it can be used
- The **type** of a variable cannot be changed after declaration
- The **value** of a variable can be changed many times

```java
int x;
int y;
int z;
x = 7;
y = 5;
z = x + y;
System.out.println(z);
```

Variables declared and initialized in single statement

```java
int x = 7;
int y = 5;
int z = x + y;
System.out.println(z);
```

Variables declared and initialized in separate statements

```java
int x;
int y;
int z = x + y;
System.out.println(z);
```

Operator Precedence

- What is the order of evaluation in the following expressions?

```
a + b + c + d + e     a + b * c - d / e

<table>
<thead>
<tr>
<th>a / (b + c) - d % e</th>
</tr>
</thead>
<tbody>
<tr>
<td>a / (b * (c + (d - e)))</td>
</tr>
</tbody>
</table>
```

Decimal Numbers

```java
float num = 5.2;
num = 1.4;
num = num * 2.0;
System.out.println(num);
```

```
2.8
```

```java
double fahrenheit = 98.6;
double celsius = (fahrenheit - 32) * 5 / 9;
System.out.println(celsius);
```

```
37.0
```
String Concatenation

```java
String course = "Data Structures";
System.out.println(course);

String s1 = "CS230 is ";
String s2 = "fun";
String fact = s1 + s2;
System.out.println(fact);

double temp = 39.2;
System.out.println("It is "+ temp + " degrees");
```

Strings in Java

- Strings in Java and Python are quite similar.
- Like Python, Java strings are immutable.
- The difference is that Java uses method calls where Python uses Operators.

<table>
<thead>
<tr>
<th>Python</th>
<th>Java</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>str[3]</td>
<td>str.charAt(3)</td>
<td>Return character in 3rd position</td>
</tr>
<tr>
<td>str[2:5]</td>
<td>str.substring(2,4)</td>
<td>Return substring from 2nd to 4th</td>
</tr>
<tr>
<td>len(str)</td>
<td>str.length()</td>
<td>Return the length of the string</td>
</tr>
<tr>
<td>str.find('x')</td>
<td>str.indexOf('x')</td>
<td>Find the first occurrence of x</td>
</tr>
<tr>
<td>str.split()</td>
<td>str.split(' ')</td>
<td>Split the string on whitespace into a list/array of strings</td>
</tr>
<tr>
<td>str.split(',')</td>
<td>str.split(',')</td>
<td>Split the string at ',' into a list/array of strings</td>
</tr>
<tr>
<td>str + str</td>
<td>str.concat(str)</td>
<td>Concatenate two strings together</td>
</tr>
<tr>
<td>str.strip()</td>
<td>str.trim()</td>
<td>Remove any whitespace at the beginning or end</td>
</tr>
</tbody>
</table>

```java
String s1 = new String("Grace Hopper");
String s2 = "CU L8R";
String s3 = ";");
System.out.println(s1.toLowerCase());
System.out.println(s1.length());
System.out.println(s2.length());
System.out.println(s2.equals(s3));
System.out.println(s2.equals("CU L8R"));
System.out.println(s2.charAt(1));
System.out.println(s1.substring(7,11));
System.out.println(s2.substring(0,2).toLowerCase());
```

Find the Errors!

```java
// This program has at least 5 errors. Can you // find them all?
public class Errors {
    public static void main(String[] args) {
        String temperature = 80.3;
        int n = 100
        n = "Wait, what?";
        print("This is fine.");
    }
}
Choose your own adventure…

* Find a group and a board

* Write your own Java program to calculate some value and print it out

* Some ideas (or make your own!):
  * Area of a circle (or other shapes) given its radius (or other necessary dimensions)
  * Volume of a box/sphere/cylinder of some given dimensions
  * Simple interest given amount, rate, time