



# Welcome to CS 149

## Introduction to Programming

Section 002

Instructor: Dr. Kevin Molloy



# Contact Info

- My JMU e-mail - [molloykp@jmu.edu](mailto:molloykp@jmu.edu)
- Class website:  
[https://w3.cs.jmu.edu/molloykp/teaching/cs149/cs149\\_2019Spring/](https://w3.cs.jmu.edu/molloykp/teaching/cs149/cs149_2019Spring/)
- My office: ISAT 216
- Office hours:
  - Tuesday 10:00 – 12:00
  - Thursday 2:15 to 4:00
  - Other times by appointment



# Weekly Routine

- Reading assignment (complete before Tuesday)
- Reading quiz in **Canvas** (before 10:00 am on Tuesday)
- Breakdown of lecture:
  - 25% lecture
  - 15% activities
  - 60% problem set solving
- Turn in problem set by Sunday at 11:00 pm to Autolab.



# Problem Sets and Projects

Each week you will be assigned up to 5 small programming activities. These activities are each short, and focus on the material we are learning that week.

We will use Autolab (<https://autolab.cs.jmu.edu>) for submission of most of the problem sets. Autolab allows you to submit your work multiple times and may give you clues as to where you need to improve your work to receive full credit.

The last few weeks of the class will focus on a larger project.



# Class Info – Getting Help

- Teaching Assistants (TAs):
  - Monday – Thursday from 5:00 pm to 11:00 pm
  - Sunday – 1:00 pm to 11:00 pm
- Piazza - post general course / assignment questions here so that peers can answer and I can certify an answer and don't have to repeat answers to multiple e-mails.
- I will generally respond to most e-mails within a 1/2 day unless it is after 8pm or a weekend (I may or may not answer e-mails until Monday morning over a weekend).



# CS149 Introduction to Programming

This class is about programming

Two parts:

- **Algorithm design** Creating a set of steps for solving some problem
- **Programming** Translating those steps into a language that a computer can execute

We will be programming in Java (version 10).



# Algorithms

There are many definitions for this term:

- [https://en.wikipedia.org/wiki/Algorithm\\_characterizations](https://en.wikipedia.org/wiki/Algorithm_characterizations)

Here is one we will use for this class:

- A series of steps for solving some problem that are detailed and clear enough that anyone following them will produce the correct output, even if they have no understanding of what the steps are supposed to accomplish.



# CS 149 Topics

Common elements of algorithms/programs:

- **Input/Output** Input comes from a user, results are reported to a user
- **Functions/Methods** Named set of instructions
- **Decisions** Some instructions only executed under certain conditions
- **Loops** Repetition of instructions
- **Variables** Named locations for storing values
- **Data Types** Categories of values that algorithms operate on “Card”, “Integer” etc.
- **Operations** Manipulation of values based on their type. E.g. integers may be added together.
- **Arrays** A sequence of related objects





# Grading

See syllabus for full grading details and breakdown, summary:

Quizzes	10%
Homework/Programming	15%
Midterm 1	20%
Midterm 2	25%
Final Exam	30%

To receive a letter grade higher than a “C”, You must get a grade of 60% of greater on the final.

This is not a weed-out course, but a **B- or higher** is required to move on to CS 159.



# Development Environment

Eclipse ([www.eclipse.org](http://www.eclipse.org)) is an IDE (integrated programming environment) that makes JAVA programming easier.

- Eclipse is installed on all lab computers and the a virtual LINUX environment (VM).
- We will now develop a Hello world program in Eclipse and upload it to Autolab.
- See [https://w3.cs.jmu.edu/molloykp/teaching/cs149/cs149\\_2019\\_Spring/EclipseSetup/](https://w3.cs.jmu.edu/molloykp/teaching/cs149/cs149_2019_Spring/EclipseSetup/) for more instructions on installing Eclipse.



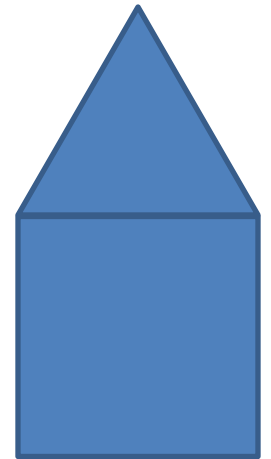
# Eclipse Useful Hotkeys

Hotkey	Action
Control-Shift F	Applying formatting rules to your source code
F11	Run your code
Control s	Saves your file



# Drawing a house

- Activity:  
Describe the 'algorithm' to draw the house shown in the picture to the right





# Drawing a House

Drawing a House: (“high level” algorithm)

1. Draw a 2” square.
2. Add a roof by drawing two line segments that begin at the top two corners of the square and meet 1” above square's center.



# Drawing a House

Drawing 2" A Square (less abstraction):

1. Place the end of your pencil in the center of a sheet of paper.
2. Draw a 2" line from left to right, parallel with the top edge of the paper
3. Without lifting your pencil, draw a 2" toward the bottom of the paper, parallel with the right edge of the paper.
4. Without lifting your pencil, draw a 2" line from right to left, parallel with the top edge of the paper.
5. Without lifting your pencil, draw a 2" line toward the top of the paper, parallel with the right edge of the paper.



# Machine Language

**Machine language** ([https://en.wikipedia.org/wiki/Machine\\_code](https://en.wikipedia.org/wiki/Machine_code)) is the lowest level operations that can be executed by the CPU(Central Processing Unit) of a computer.

What does this program do?

169 1 160 0 153 0 128 153 0 129 153 130 153 0 131 200 208 241 96

Examples:

- ARM – Probably on your phone(Android)
- X86 – Probably on your computer



# Machine Language

Examples of individual machine Instructions are:

- Perform an arithmetic operation
- Change the contents of memory
- Change the next instruction to fetch
  - Jump backwards → iteration
  - Jump forwards → conditional execution
- Generate output
- Obtain input





# Machine Language

## Cons:

- Hard to work with
- Not portable – different across platforms (different CPU can have different instruction sets) and operating systems (Mac, Windows)

## Pros (at least for us):

- Few people *need* write in machine language (so odds are, you will not need to do this)
- We program in Java, C, C++, Python, Ruby, Go, php, Perl, etc...

# Hello World- Java Program

A class is a collection of methods

```
public class Hello {
```

This is the main method

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

```
        // generate some simple output
```

```
        System.out.println("Hello, World!");
```

```
    }
```

```
}
```



*Token = identifier | keyword | separator | operator | literal | comment*

1. **Identifiers:** names the programmer chooses
2. **Keywords:** names already in the programming language
3. **Separators** (also known as punctuators): punctuation characters and paired-delimiters
4. **Operators:** symbols that operate on arguments and produce results
5. **Literals** (specified by their **type**)
  - Numeric: **int** and **double**
  - Logical: **boolean**
  - Textual: **char** and **String**
  - Reference: **null**
6. **Comments**

# To Do's

**ASAP:** Take the Course Intro Survey if you haven't already

By Thursday's class:

- Acquire the book (free pdf download online) and read Chapter 1 *Thinking in Java*.
- Complete the Reading Quiz in Canvas (**due Thursday**)
- Complete Problem Set 1 by Sunday

Optional:

- Install Java (version 10) + jGRASP on your own computer and/or install Virtualbox and Linux Mint 19 from the Virtual Machine on your computer.

