

# CS 1: Welcome and Introduction

---

CS 163 and CS 164: Java Programming

WARM UP QUESTION – Name, Why are you in this course, and something you are most nervous about?

Colorado State University  
Computer Science Department

Slides Originally Created by Albert Lionelle (Albert.Lionelle@colostate.edu)



Colorado State University

# Weekly Announcements!

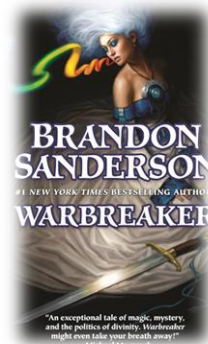
- Will have these up every week!
- Start of every class as you come to class.
- Class FRIDAY
  - Reading 1 done **before** Friday
  - iClicker Setup by Friday
  - Lab Thursday is to help with that
- Always, always plan for readings to be done before class.

## TODO Reminders:

- Setup MS Teams
- Reading 1 (Zybooks)
- Syllabus Quiz
- Knowledge Check

# Who Am I?

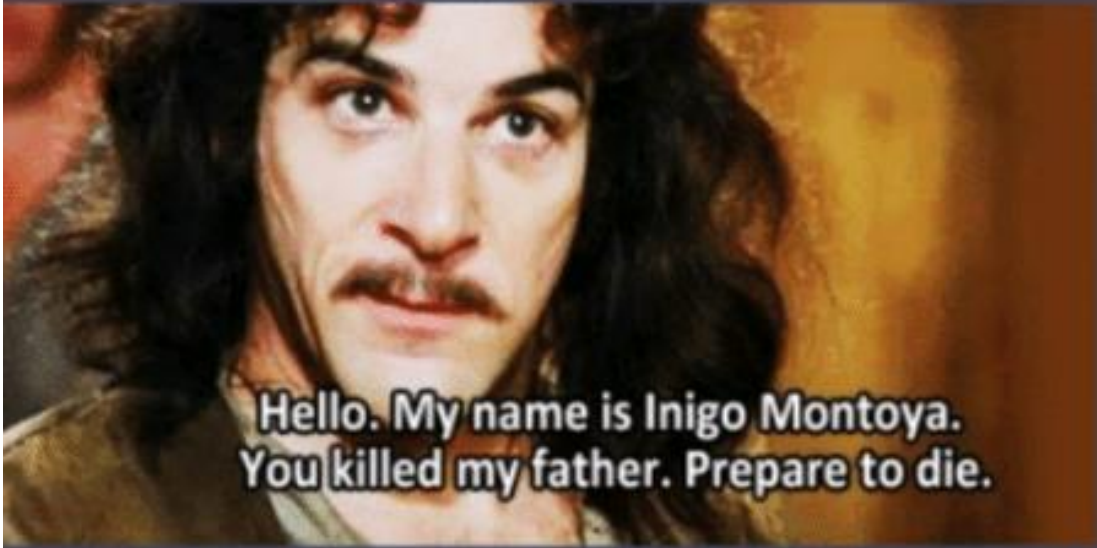
- Masters of Science in Computer Science - From CSU!
  - Studied Human Computer Interaction (HCI) working on detecting déjà vu.
- Research Interests
  - Computer Science Education
  - Brain Computer Interfacing
  - Computer vision
  - Computer Languages
- Work experience
  - 3<sup>rd</sup> Year Instructor
  - 4 Years work as both a UTA and a GTA for many course in the department
- Outside Interests
  - Reading
  - Gaming
  - Tabletop
  - Roleplaying games
  - Archery
  - Painting





# Instructor: How to Contact Me?

- MS Teams Private Message
  - \*BEST WAY\*
  - email (poor), canvas (very bad!!)
- General questions
  - Post in the general channel!
  - Let's other see the answer
- Check the Syllabus for Office Hours!
  - Mixed MS Teams and Office
    - stop by either but may need to wait.



Remember Inigo Montoya:

1. Polite greeting
2. Name
3. Relevant personal link
4. Manage expectations

# Remember!

- Remind me which class! (I teach multiple)
  - Relative personal link == class you are in ☺
- Manage Expectations
  - Short question

# Topics Covered

- Basic Java Programming
  - Variables
  - Control Structures (Branching)
  - Loops
  - Arrays
  - ArrayLists
  - Classes
  - Inheritance
  - Polymorphism
- Essentially - learning the tool to build basic programs!
- Problem Solving
- Divide-Conquer-Glue
  - A way to look at the world



Like learning music - programming takes practice, practice, practice



Attribution: [Algont](#)



## Computational Thinking:

Decomposition

Pattern Recognition

Abstraction

Algorithms

# Teaching Approach/Concepts

- Based on Psychology of Learning (4 week cycles)
  - Spacing
  - Interleaving
  - Practiced Recall
  - Elaboration
  - Reflection
- Grading
  - Formative – Can be redone!
  - Summative – Demonstrate what you know
- You are learning
  - A new language
  - A different way of thinking (Divide-Conquer-Glue)
  - OK to struggle!

To have another language is  
to possess a second soul. -

Charlemagne (748 –814)

# Course Structure - Follow Canvas

- Readings
  - Due before Lectures - Sunday, Tuesday and Thursday nights
- Lectures
  - Active learning, a lot of group discussion and coding – BE HERE – or you will miss out.
- Labs
  - Meant to be done after lecture content, due day assigned - coding/writing code
- Knowledge Checks
  - Required to move onto the next module
  - Your best study tools
- Exams
  - Canvas exams – reading content
  - Coding exams – writing content
- Practical
  - These are large and *hard* programming assignments (usually 2 weeks), that bring it all together



# Knowledge Checks

- Focus on **Reading Code**
- Also help with recall
  - Best thing to do – go back to them
  - Interleave – pick different orders to redo them from time to time
  - Spacing – do some every night!
  - Psychology of learning – this helps!
- Best way to study for exams?
  - Every night – knowledge checks, practice exam
    - Retesting + spacing
  - You can do them 100 times, get different results every time!
    - Highest result is the one kept

	ReStudy	ReTesting
Massed	Most People	
Spaced		Ideal For Recall

# Labs

- Tuesday Labs
  - 2 of 4 grade is based on activities / participation
    - Self-explanation in comments, drawing out graphs, writing tests
  - 2 or 4 grade is based on submitted / auto graded (one is often just debugging)
  - All is meant to be done during lab.
  - Miss a lab, you need to coordinate with the lab TA.
- Thursday Labs
  - Meant to building small programs
  - Can often take more than one day
  - Will have both provided code and your own code you write
  - 4 of 4 is auto graded / based on working code.
  - Some of them \*build\* on each other, so you need past ones completed!

# Help Sessions

- Sign up for a help session!
- Weekly meetings lead by TAs
- Go over:
  - Course content
  - Assignments
  - Additional related material
- Can't make it
  - Online help session

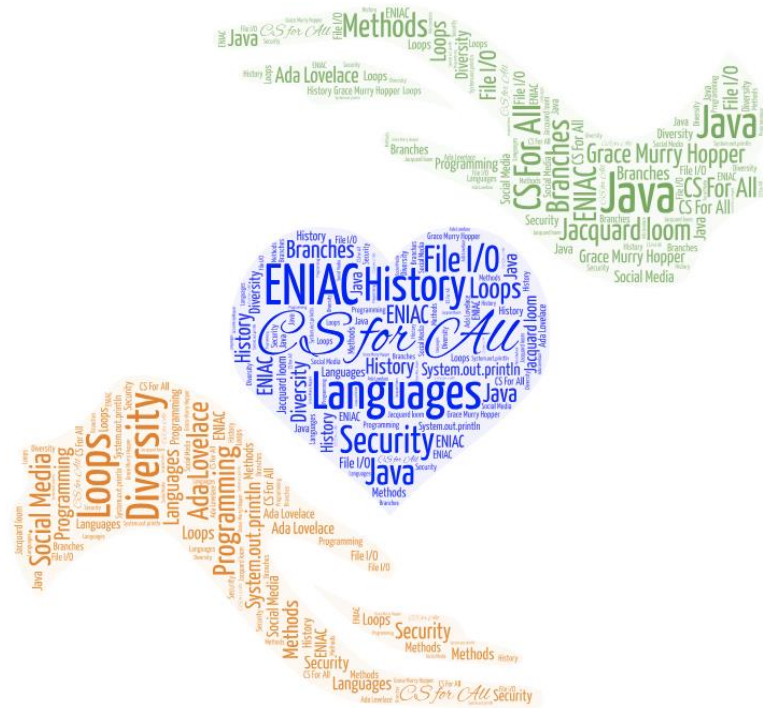
# MS Teams!

- MS Teams
  - Used in industry
  - We make use for the course (mixed with online and on-campus students)
- Install the Application! (not just the browser)
- Use the General Channel (Study Group!)
  - General Questions
  - Knowledge Checks
  - Reading
  - aka, any question that doesn't require posting code \*you\* write

# Most Importantly

- Remember to Ask Questions
- Stay Ahead / Don't Fall Behind
- Be kind to yourself - no one gets everything the first time.
- Have fun!

Computer Science is using technology to solve problems!



And who was the first programmer?

## The Right Honourable Countess of Lovelace



Photo By:  
Alfred Edward Chalon [Public domain], via Wikimedia Commons

- [Ada Lovelace](#) (1815-1852)
- English mathematician who worked with [Charles Babbage](#) on his [Analytical Engine](#)
- In 1843 she translated an article written by the Italian mathematician and engineer Luigi Federico Menabrea, "Notions sur la machine analytique de Charles Babbage" and supplemented it with her own "Notes"
- Her "Notes" contains what many consider to be the first computer program
- Ability to connect the Arts and Science, she developed a vision of the capabilities of computers to go beyond calculation, it can do anything that can be noted in symbols, including words and music (<https://www.britannica.com/biography/Ada-Lovelace/images-videos>)
- Ada Lovelace Day – second Tuesday in October