CSCI 141:
Computer Programming I

Lecture 0
Introduction, Logistics, Hello World
Today

- What is this course about?
- Why are we here?
- Who is this character?
- Some course logistics
- Let’s write some code already!
What is this course about?

• What is this course about?

• What will you learn?
What is this course about?

From the course catalog:
Basic concepts of computer programming using an object oriented programming language. Topics covered: introduction to the development environment, introduction to algorithms, elements of a programming language, including data types, packages, control structures, procedures and functions, basic input and output, arrays and records, text files, strings, variant records. Algorithm development, problem solving and software engineering are emphasized.
What is this course about?

From the course catalog:
Basic concepts of computer programming using an object oriented programming language. Topics covered: introduction to the development environment, introduction to algorithms, elements of a programming language, including data types, packages, control structures, procedures and functions, basic input and output, arrays and records, text files, strings, variant records. Algorithm development, problem solving and software engineering are emphasized.
What is this course about?

Computer Programming:

• data types
• control structures
• functions
• strings
• arrays

These are the “nuts and bolts”: the syntax and semantics of programming languages.
What is this course about?
From the course catalog:
Basic concepts of computer programming using an object oriented programming language. Topics covered: introduction to the development environment, introduction to algorithms, elements of a programming language, including data types, packages, control structures, procedures and functions, basic input and output, arrays and records, text files, strings, variant records. Algorithm development, problem solving and software engineering are emphasized.
What is this course about?

Problem Solving and Software Engineering:

- Break down and analyze problems
- Design algorithms that solve problems
- Describe algorithms in pseudocode
- Implement algorithms using clearly written, correct Python code.
- Fix errors and make changes to the code once it’s written.

Computer Programming:

- data types
- control structures
- functions
- strings
- arrays
Why learn to program?

• Why do you want to learn how to program?
Why learn to program?

• Some ideas:

• Get a job with cool perks and a high salary
Why learn to program?

• Some ideas:
  • Get a job with cool perks and a high salary
  • Automate repetitive tasks
Hi there,

Ice cream for the first person who can get me what I want.
Count the number of times each person's name occurs with Column A=Person's Name, Column B=# of Occurrences.
READY? GO!

Thanks,
Andrew

Cornell Outdoor Education
Climbing Program Coordinator
Why learn to program?

• Some ideas:
  • Get a job with cool perks and a high salary
  • Automate repetitive tasks
  • Process or analyze data you encounter in your chosen profession
  • Execute your creative vision
  • Understand what's going on inside the computers you use daily
  • Make friends with our future robot overlords
Who is this character?
Computer Vision: Familiar Examples

- In-Camera Face Detection
- Autonomous Driving
- Panorama Stitching
- Image Search
Video Spectrograms for Exploration and Discovery in Long Fixed-Camera Video Streams

Category: Research
Paper Type: algorithm/technique
Overhead Imagery Analysis

Count birds!

Measure international border legibility!
Logistics

The syllabus is [on] the course webpage:

https://facultyweb.cs.wwu.edu/~wehrwes/courses/csci141_21s/

This link can also be found on the Syllabus page on Canvas.

CSCI 141 - Computer Programming I
Scott Wehrwein
Spring 2021

- Course Overview
- Assessment
- Resources for Getting Help and Support
- Logistics
- Schedule
- Course Policies

Course Overview

- Syllabus and Course Website (you are here): https://facultyweb.cs.wwu.edu/~wehrwes/courses/csci141_21s
Syllabus Highlights

- Flipped structure
- Assessment components
- Labs
- Schedule table
- Slip days
- Academic honesty
Your Tasks for Friday
(in brief; I will send details in an announcement by tonight)

• Read the syllabus

• Watch the lecture videos for Friday

• Work through Friday's Exercises

• Get on the class Discord server

• Find your group number
About You Survey

Survey on Canvas - fill out by Monday

• Q1: How many quarters have you been at Western?

• Q2: How many months of programming experience do you have?

• Q3: Are you planning to major in CS?

• Q4: Why do you want to take this class?

• Q5: Name one activity you enjoy outside of school.
My Expectations

Q2: How many months of programming experience do you have?

0

Some of you have prior experience, but zero is what I will assume. We'll look at the survey results and talk more about this next time.
Let’s write some code already

- **Python** is our chosen programming language in this course.

- A *programming language* is a language a computer can “understand” and execute (more on what this means next time)

- We’ll use a program called **Thonny** to write our Python code.

- Thonny is an example of an “*Integrated Development Environment*” (IDE): a program that provides all the features you need to write, run, and fix errors in programs.
Without further ado... Hello, world!
Hello, world!

- hello.py

- Concepts demonstrated:
  - Basic usage of Thonny
  - Comments
  - Print function
  - Single and double quoted strings