Introduction to Computer Systems

Contents

Resources

Syllabi

This new course links together different ideas that you have encountered but not covered deeply in other courses. We'll learn about tools used in programming and how they work. The goal of this course is to help you understand how your computer and programming environment work so that you can debug and learn independently more confident.

Quick Facts

- **Course time:** TuTh 12:30PM - 1:45PM
- **Credits:** 4

To request a permission number [complete this google form](https://forms.gle/4Le3bZtE3VWmbz2G9

you must be signed into your URI google account to access the form

Why Take this course

1. use and understand git/ GitHub
2. make sense of cryptic compiler messages
3. understand how file organization impacts programming
4. fulfill your 300 level CSC elective requirement
5. preview ideas that will be explored in depth in 411 & 412
6. Further your understanding of systems abstractions

Topics covered

this is a partial list

- git and other version control
- bash and other shell scripting
- filesystems
- basics of hardware
- what happens when you compile code
- what are the different types of software on your computer

Catalog Description

How the history and context of computing impacts the practice of computing today. Tools used in programming and computational problem solving. How programming works from high level languages to hardware. Survey of computer hardware and representation of information. **Pre:** CSC110, any 200 level CSC course, or equivalent (prior experience programming in at least two languages).

Learning Outcomes

By the end of the semester, students will be able to:

1. Differentiate the different classes of tools used in computer science in terms of their features, roles, and how they interact and justify positions and preferences among popular tools
2. Identify the computational pipeline from hardware to high level programming language
3. Discuss implications of choices across levels of abstraction
4. Describe the context under which essential components of computing systems were developed and explain the impact of that context on the systems.

FAQ

What will the course be like?

Does the course focus on going in depth on a few topics or covering a variety of topics with less depth?

How many credits is this class?

Why does e-campus let us choose a number of credits?

How does this count toward my CS Major?

References

These resources are available to students. Level 1 is a basic scratch the surface explanation of the topic. Level 2 is an intermediate level of explanation. Level 3 is an in-depth explanation of the topic.

History of Computers

Tools of the Craft

Survey of Hardware Components

What is a CPU, and What Does It Do?	**1**	**Article**	**Easy to read article that explains CPUs and their use. Also touches on “buses” and GPUs.**
Processors Explained for Beginners	1	Video	Video that explains what CPUs are and how they work and are assembled.
The Central Processing Unit	1	Video	Video by Crash Course that explains what the Central Processing Unit (CPU) is and how it works.
Input/Output Device	1	Article	Easy to read article that explains what an I/O device is. Touches on troubleshooting I/O Errors and examples of I/O devices.
Computer Bus	1	Article	Article that explains what computer buses are and how they work. Talks about the different types of buses and bus speeds.
What is a GPU?	1	Article	Defines a GPU in simple terms. Explains the difference between a CPU and a GPU. Also touches on how it works and its uses.
What Are Parallel Processing Systems	2	Article	Explains what a parallel processing system is and how it works.
Logic Gates	1	Video	Crash Course video that explains boolean logic and their respective gates in a visual and easy to understand way.
[What are derived logic gates?] (https://www.youtube.com/watch?v=wGilEPBfcT8	1	Video	Video that explains what the derived logic gates are (NAND, NOR, XOR, XNOR). Uses simple, easy to understand real-world examples such as light switches in a room.

What is a CPU, and What Does It Do?	**1**	**Article**	**Easy to read article that explains CPUs and their use. Also touches on “buses” and GPUs.**
Registers and RAM	1	Video	Crash Course video that explains what registers are and how RAM works.

Software Infrastructure

Operating Systems: Crash Course	**1**	**Video**	**Very basic introduction to operating systems and the history of how they began.**
Files and File Systems	1	Video	Crash Course video that gives a very basic introduction to files and file systems.
Abstraction Layers Explained	1	Video	Video that gives the basics of how abstraction layers are organized.
The Linux File System Explained	1	Video	Video that explains the Linux file system. Explains by showing.
Programs, Processes, and Threads	1	Article	Article that explains the differences between programs, processes, and threads. Has helpful diagrams that show the differences and how each one works.
Cache Memory in Computer Organization	1	Article	Explains how cache memory works and how it is accessed by the CPU. Also talks about cache mapping, types of cache, and cache performance.

Number Systems

Representing Numbers and Letters with Binary	**1**	**Video**	**Basic intro to how numbers and letters are represented using binary.**
Binary, Octal, and Hexadecimal	1	Text	Explains the three different types of number representations used in computer memory. Explains which ones are more efficient for storing information.
Numeral Systems	1	Text w/ chart	BRIEFLY explains how each numeral system works (binary, octal, decimal, and hex). Shows examples of each numeral system would work. Has a conversion table at the bottom of the page.

Machine Representation of Data

What is a Bitwise Operator and How to Use Them	**2**	**Video**	**Presentation that explains what bitwise operators are and how they work. The presentor gives examples of their use and shows them in action.**
Why Your Storage or RAM Size Doesn't Add Up	1	Article	Talks about the difference between base-10 and base-2 storage options. Table that visualizes the differences between the two number systems.
Integer and Floating-Point Number Representation	2	Text	3.0-3.10 for information on integer representation and 1's and 2's complement. 4.0-4.4 for information and exercises on Floating-Point Numbers.
What are Overflow and Underflow?	1	Article	Explains what overflow and underflow are. Each is explained using an example.

Why take this class

You should take this class if you want to be introduced to some topics prior to taking 411 and 412. This class also allows you to gain some experience using the terminal, understanding the term `computing` and the topics surrounding it, and using version control systems such as git which are useful in the professional world.

This class offers great experience with real world computer science tools like github. This class will also expose you to intermediate computer science concepts. In total the skills learned in this class will make you a much more rounded computer science student.

You should take this class if you want to know more about what your computer is doing. This class will help you better understand error messages and will teach you bash scripting and git. I have not taken 411 yet but I hear this class is good to take before 411. I also believe this class is good to take before 305 because we use git a lot in that class.

You should take this class if you want to get a headstart on your education and career. This class covers quite a bit, but each topic is designed to give you just enough so that you're not completely in the dark when you have to learn this stuff for a class. Additionally, being comfortable with version control systems is essential for not only school projects, but also your internships and jobs.

By Sarah M Brown
© Copyright 2022.