Overview

This document is an attempt to build a shared understanding of topics expected to be covered in CS160 as defined in the Computer Science Major Transfer Map.

We have identified six core pillars of content - Computational Thinking, Programming, Motivation, Foundations of Computing, Topics in CS, and Ethics & Society. We expect that any version of CS160 will include all six pillars, but that there can and should be variance in how they are covered.

Within each pillar potential topics are defined and for each topic, one or more examples of what might be covered. Some pillars (Computational Thinking, Motivation, Ethics & Society) have a few specific topics. In these areas, there are a few key ideas deemed essential, although those ideas could be developed in many different ways. In other pillars (particularily Topics in CS) there are a large number of potential topics suggested. For those, coverage of every topic is not essential - the topics should be considered more as a list of suggestions from which to draw from while creating an implementaiton of CS160.

Coverage of the pillars can and should overlap. Computational Thinking and Programming naturally go hand in hand. There are topics that have aspects relevent to both Foundations of Computation and Topics in CS. Many Topics in CS are natural places to discuss concerns related to Ethics & Society. And, hopefully, Programming, Ethics & Society, and Topics in CS all serve to support the Motivation pillar.

Below the pillars are described and appoximate time budgets suggested. The Content tab lists actual topics and examples for each pillar. To emphasize the flexibility intended in the Topics in CS category, a menu of suggested topics is presented in the Topics in CS page as a starter menu of ideas that might be covered.

Pillar	*What students should get from this pillar*	**Approximate Time Budget**
Computational Thinking	Develop an understanding of how to solve problems with algorithms.	50% combined
Programming	Become familiar with the tools used to implement algorithms. The goals for all programming topics are more exposure than mastery - students going on in CS will develop programming proficency in CS161/162.	
Foundations of Computation	Learn the basics of how computers function. Develop a base of knowledge to understand topics encountered in CS161/162/260. (The earth from which the trees in CS161/162 grow)	20%
Topics in CS	Exposure to topics studied in the field of computer science. What kinds of things might students study in Junior or Senior years? (The forest to go with CS161/162's trees)	20%
Ethics & Society	Consider the impacts of computation on people and society and ethical questions that arise from these impacts.	5% Independent (and Integrated with other topics)
Motivation	Help prospective majors develop a sense of competency and belonging within the field. Inspire non-majors to consider computer science as a major or just appreciate its value to them.	5% Independent (and Integrated with other topics)

Misc Notes

- Computer literacy should be considered a prereq for 160

- The AP CS Principles course has been identified as satisfying CS160 within the CS Major Transfer Map. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-conceptual-framework-2020-21.pdf

Contributors

Stephanie Allen	PCC
Hank Childs	UO
Lucas Cordova	WOU
Karla Fant	PSU
Kathleen Freeman	UO
Ellie Harmon	PSU
Gayathri Iyer	PCC
Becka Morgan	WOU
Troy Lanning	Klamath CC
Andrew Scholer	Chemeketa CC
Robert Surton	Chemeketa CC
Ken Swartwout	COCC

Pillar	Topic	Example
Computational Thinking	Problem Solving	Reading and writing specifications
Computational Thinking	Problem Solving	Using or designing Flowcharts / Process Diagrams
Computational Thinking	Problem Solving	Designing test cases for algorithms/functions
Computational Thinking	Problem Solving	Decomposing problems
Computational Thinking	Problem Solving	Structured problem solving - working from a description, to specifications, to computational process, to snippets, to program, to testing
Programming	Understand and Use Existing Code	Interact with visualizations of code execution and/or mentally simulate results of steps
Programming	Understand and Use Existing Code	Complete or fix existing code samples by filling in blanks or doing Parsons problems
Programming	Understand and Use Existing Code	Modify existing programs - tweak algorithms, adjust parameters
Programming	Understand and Use Existing Code	Use code libraries to perform interesting tasks (e.g. draw graphics)
Programming	Using variables	Use variables to store different types of information (numbers, strings, other provided data types) and then manipulate those values. (Assignment and expression evaluation)
Programming	Using variables	Scope as applied to variables - local vs global
Programming	IO	Read information into a program and produce output (console, file, etc...)
Programming	Functions	Writing functions to use as abstractions within other code
Programming	Functions	Passing information to/from functions via parameters and return values
Programming	Control Structures	Using conditional logic to selectively execute statements
Programming	Control Structures	Complex logic using and, or, not, and nested conditionals
Programming	Control Structures	Loops to avoid repetiton and flexibly repeat statements
Programming	Complex Data	Strings as sequences (indexing into, getting substrings, etc...)
Programming	Complex Data	Use of lists, dictionaries, or other collections of data as a consumer
Programming	Practices	Documentation for code
Programming	Practices	Debugging - the process of finding and fixing errors in code
Programming	Practices	Iterative development of code
Programming	Practices	Review the programs of peers and give feedback
Programming	Practices	Working with others on code: pair programming or work in a group to implement parts of a program
Foundations of Computation	Digital Information	How computers represent information: bits/bytes
Foundations of Computation	Digital Information	Binary numbers (simple positive integers)
Foundations of Computation	Digital Information	Hexadecimal as a way to represent large binary numbers
Foundations of Computation	Digital Information	Representing data like text and digital images with bits
Foundations of Computation	Computation as a Series of Abstractions	Importance of abstraction stacks (e.g. transitors/logic gates/circuits/CPU or electricity/bits/RGB color or IP/TCP/application or machine/assembly/high level code)
Foundations of Computation	Historical Computing	Origins of computing
Foundations of Computation	Historical Computing	Evolution of computing; Moore's law
Foundations of Computation	Computing Hardware	Identify transistors/switches as the basic building block of processing.
Foundations of Computation	Computing Hardware	Given inputs, predict the output of a logic circuit.
Foundations of Computation	Computing Hardware	Machine code and how high level programs become machine code (at a very abstract level)
Foundations of Computation	Computing Hardware	Major components of a computer (CPU, RAM, Secondary Memory) and the role they play
Foundations of Computation	Computing Hardware	Role of the operating system in providing abstractions for hardware and allocating scarce resources

Pillar	Topic	Example
Topics in CS	Variable...	There are a wide variety of topics which might be coved. Any topic covered in later CS courses, or any application of computation to other domains, fits in this pilar. Some courses might cover a wide variety of topics at a basic level while other courses might dive into fewer topics more deeply (e.g. CS160 with a particular focus in Data Science, or Software Engineering, etc...) For a list of topic ideas, see the Topics in CS page.
Ethics & Society	Ethics & Rights	Copyright, Open Source, and Creative Commons
Ethics & Society	Ethics & Rights	Risks to privacy from using modern technology
Ethics & Society	Ethics & Rights	Security risks and their impacts - the costs of poorly designed and implemented software
Ethics & Society	Ethics & Rights	Societal impacts of algorithms like search/filtering, machine learning, etc...
Ethics & Society	Ethics & Rights	Ways in which algorithms are not "neutral" - how they can have disparate effects on different groups and individuals
Motivation	Identify with Computation	Computing as a potential future pursuit for students - either as a major or as a topic supporting other interests
Motivation	Identify with Computation	Programs as creative expression
Motivation	Identify with Computation	Programs to answer questions/solve problems
Motivation	Degree & Career Awareness	Career opportunities in CS related fields
Motivation	Degree & Career Awareness	How computation has changed a non-CS field
Motivation	Degree & Career Awareness	Educational pathways in CS related disciplines. CS/SE/IS/IT/CE/etc...

Pillar	Topic	Example
Topics in CS	Cybersecurity	Security risks and their impact on society
Topics in CS	Cybersecurity	Multi-factor authentication
Topics in CS	Cybersecurity	Encryption and privacy
Topics in CS	Algorithms	Limitations of algorithms. Identifying that some problems that are easily solved, some are so hard as to be practically unsolvable, and some are unsolvable.
Topics in CS	Algorithms	Different algorithms can be used solve the same task and they will often have tradeoffs (linear vs binary search)
Topics in CS	Algorithms	Efficency is typically quantified with a function that relates the input size to the amount of work required to process that input.
Topics in CS	Algorithms	Various "real world" algorithms. PageRank, Encryption, Machine Learning, Compression, etc...
Topics in CS	Parallel Computation	Multi-core computers and networked computation
Topics in CS	Parallel Computation	Challenges associated with parallel computation
Topics in CS	The Internet	Design philosophy of the internet - decentralized and robust system for finding and communicating with other machines
Topics in CS	The Internet	How the internet model impacts privacy and security
Topics in CS	The Internet	Protocol layers are involved in an internet request
Topics in CS	The Internet	Privacy on the internet: VPNs, TOR, etc...
Topics in CS	AI	Overview of AI as an attempt to produce intelligent looking behavior
Topics in CS	AI	Machine learning and the modern explosion in applications of AI
Topics in CS	AI	Interact with one or more AI toolkits
Topics in CS	Data	Use data scientist tools (i.e. Google Colab Notebooks, Jupyter Notebooks) to collaborate, code, and visualize datasets.
Topics in CS	Data	Create charts and plots to visualize data with which to draw conclusions by using quantitative reasoning.
Topics in CS	Data	Basic relational database representation and queries
Topics in CS	Data	Linking of data tables using keys and joining them to assemble data
Topics in CS	Applications in law and legal reasoning	Applying law to new technology and new technology to law
Topics in CS	Digital Humanities	Intersection of computing technologies and the humanities
Topics in CS	Applications in philosophy	AI, robotics, modeling, ethics
Topics in CS	Graphics	Rasterization vs raytracing -- how they work and comparing the two from a Big O perspective
Topics in CS	Graphics	Myriad optimizations for improving performance, like spatial search data structures
Topics in CS	Graphics	How to use a GPU to do graphics -- APIs (OpenGL, Vulkan) plus the amazing software stack that enables shader programs. Also: what is a GPU?