CROSSWALK

Praxis 5652 Computer Science Competencies

Foundations of Computer Science - Praxis Prep

The Foundations of Computer Science - Praxis Prep course addresses all 196 competencies in the 5 main units and 56 sub-topics of the Praxis 5652 list.

In the left column are the **Praxis** elements, the right the corresponding Foundations units.

In several instances, liberties were taken to reorder and/or combine topics into a more cohesive and logically sequential learning unit. For instance, Module 3 is divided into three parts due to the volume of material, and several topics were grouped together in a modified sequence to better provide a learning sequence.

WeTeach_CS

I. Impacts of Computing

Module 1 - Impacts of Computing

1A1.	**Understand computing as a way of expressing creativity, solving problems, enabling communication, and fostering innovation in a variety of fields and careers**	1A1: Creativity and Innovation in Computing
1A2	**Know the obstacles to equal access to computing among different groups and the impact of those obstacles**	1A2: Obstacles to Equal Access
1A3	**Understand beneficial and harmful effects of computing innovations and the trade-offs between them**	1A3: Computing Innovations - Benefits and Tradeoffs
1B1	**Know different methods of protecting intellectual property rights and the trade-offs between them in a variety of contexts (e.g., Creative Commons, open source, copyright)**	1B1: Intellectual Property Issues
1B2	**Understand ethical and unethical computing practices and their social, economic, and cultural implications**	1B2: Ethics of Computing

1B3	**Know privacy and security issues regarding the acquisition, use, and disclosure of information in a digital world**	1B3: Digital Privacy and Security
II. Algorithms and Computational Thinking		**Module 2 - Algorithms and Computational Thinking**
2A1	**Understand abstraction as a foundation of computer science**	2A1: Abstraction
2A2	**Know how to use pattern recognition, problem decomposition, and abstraction to develop an algorithm**	2A2: Developing Algorithms
2A3	**Understand number base conversion and binary, decimal, and hexadecimal number systems**	2A3: Computer Number Bases
2A4	**Understand how to develop and analyze algorithms expressed in multiple formats (e.g., natural language, flowcharts, pseudocode)**	2A4: Pseudocode, Flowcharts
2B1	**Be familiar with the limitations of computing in terms of time, space, and solvability as well as with the use of heuristic solutions that can address these limitations**	2B1: Space/Time Limitations and Heuristics
2B2	**Understand searching and sorting algorithms; can analyze sorting algorithms for correctness and can analyze searching algorithms for correctness and efficiency**	2B2: Searching and Sorting Algorithms
2B3	**Understand simple recursive algorithms (e.g., n factorial, sum of first n integers)**	2B3: Recursive Algorithms
2B4	**Be familiar with the use of randomization in computing**	2B4: Randomization
III. Programming		**Module 3 - Programming**
3A1	**Understand how to write and modify computer programs in a text-based programming language**	3A1, A2, A4, B3, B5, B6: Programmming Topics
3A2	**Understand how to analyze computer programs in terms of correctness**	3A1, A2, A4, B3, B5, B6: Programmming Topics
3A3	**Know the concepts of extensibility, modifiability, and reusability**	3A3: Extensibility, Modifiability, Reusability
3A4	**Understand the three basic constructs used in programming: sequence, selection, and iteration**	3A1, A2, A4, B3, B5, B6: Programmming Topics

3A5	**Understand how to use standard operators (i.e., assignment, arithmetic, relational, logical) and operator precedence to write programs**	3A5: Standard Operators
3A6	Understand how to use variables and a variety of data types	3A6: Part One - Data Types and Variables 3A6: Part Two - Procedures, Parameters, Arrays, Lists, Data Structures
3B1	**Understand how to write and call procedures with parameters and return values**	3B1: Procedures and Parameters
3B2	**Know the concepts of event-driven programs that respond to external events (e.g., sensors, messages, clicks)**	3B2: Event-Driven Programming
3B3	**Be familiar with usability and user experience (e.g., ease of use and accessibility)**	3A1, A2, A4, B3, B5, B6: Programmming Topics
3B4	**Be familiar with dictionaries/maps, stacks, and queues**	3B4: Stacks, Queues and Maps/Dictionaries
3B5	**Understand how to use debugging techniques and appropriate test cases**	3A1, A2, A4, B3, B5, B6: Programmming Topics
3B6	**Be familiar with characteristics of well-documented computer programs that are usable, readable, and modular**	3A1, A2, A4, B3, B5, B6: Programmming Topics
3B7	**Be familiar with techniques to obtain and use feedback to produce high-quality code (e.g., code reviews, peer feedback, end user feedback)**	3B7: Feedback
3B8	**Know how to use libraries and APIs**	3B8: Using Libraries and APIs
3B9	**Understand programming techniques to validate correct input and detect incorrect input**	3B9: Input Validation
3B10	**Be familiar with the features and capabilities of integrated development environments (IDEs)**	3B10: Using IDEs
3B11	**Be familiar with the differences between low- and high-level programming languages**	3B11,12,14: Programming Language Concepts
3B12	**Be familiar with different programming paradigms**	3B11,12,14: Programming Language Concepts
3B13	**Know object-oriented programming concepts**	3B13: Object-Oriented Programming Concepts

3B14	Be familiar with program compilation and program interpretation	3B11,12,14: Programming Language Concepts
IV. Data		**Module 4 - Data**
4A1	Understand bits as the universal medium for expressing digital information	4A1: Binary Digits
4A2	Be familiar with concepts of data encryption and decryption	4A2: Encryption/Decryption
4A3	Know how to use computational tools, including spreadsheets, to analyze data in order to discover, explain, and visualize patterns, connections, and trends	4A3: Computational Tools
4B1	Be familiar with the use of computing in simulation and modeling	4B1: Simulation/Modeling
4B2	Be familiar with methods to store, manage, and manipulate data	4B2: Data Storage and Management
4B3	Be familiar with a variety of computational methods for data collection, aggregation, and generation	4B3: Computational Methods
V. Computing Systems and Networks		**Module 5 - Computing Systems and Networks**
5A1	Know that operating systems are programs that control and coordinate interactions between hardware and software components	5A1: Operating Systems/Hardware/Software
5A2	Be familiar with computing systems embedded in everyday objects (e.g., Internet of Things [IoT], ATMs, medical devices)	5A2: Embedded Systems and IoT
5A3	Know the capabilities, features, and uses of different types of computing systems (e.g., desktop, mobile, cluster)	5A3: Computing Systems
5A4	Be familiar with computers as layers of abstraction from hardware (e.g., logic gates, chips) to software (e.g., system software, applications)	5A4: Hardware/Software/Layers of Abstractions
5A5	Be familiar with the steps required to execute a computer program (fetch-decode-execute cycles)	5A5: Fetch/Decode/Execute Cycle
5A6	Be familiar with trade-offs between local, network, and cloud computing and storage	5A6: Computing and Storage Issues
5A7	Be familiar with communication between devices	5A7: Network Communication

5B1	**Know components of networks**	5B1,2 and 3: Network Components, Functionality, Protocols
5B2	**Be familiar with factors that have an impact on network functionality**	5B1,2 and 3: Network Components, Functionality, Protocols
5B3	**Be familiar with how Internet and Web protocols work**	5B1,2 and 3: Network Components, Functionality, Protocols
5B4	**Be familiar with digital and physical strategies for maintaining security**	5B4: Network Security Strategies
5B5	**Be familiar with concepts of cybersecurity**	5B5: Five Pillars of Cybersecurity
5B6	**Be familiar with the components that make up the Web (e.g., HTTP, HTML, browsers, servers, clients)**	5B6: WWW Components/Protocols