

Exploring Computer Science (2025)

Unit 1: Orientation and Ongoing Skills 1.1

1 Identify school policies and safety procedures related to Exploring Computer Science (ECS). 1.1

- a Examine the school handbook, acceptable-use policy for technology, and safety procedures. 1.1.A
 - b Preview the course outline and its relevance in today's workforce. 1.1.B
 - c Recognize appropriate safety measures related to technology in the computer lab and online safety. 1.1.C
-

2 Investigate using an online learning management system (LMS). DOK2 1.2

- a Discover online learning environments and how they operate among teachers and students. 1.2.A
 - b Demonstrate proper email etiquette. 1.2.B
 - c Participate in online learning methods (discussion boards, student journals, blogs, wikis, and so forth). 1.2.C
 - d Collaborate with teachers and peers through an online system. 1.2.D
-

3 Recognize opportunities to participate in student organizations related to technology and computer science. DOK1 1.3

- a Identify student organizations available at the school for technology and computer science. 1.3.A
 - b List student competitions available through each organization. 1.3.B
-

4 Demonstrate knowledge of 21st-century skills. DOK2 1.4

- a Demonstrate effective collaboration and teamwork. 1.4.A
 - b Demonstrate creativity and imagination. 1.4.B
 - c Utilize critical thinking through effective reasoning, making judgments and decisions using journaling. 1.4.C
-

5 Demonstrate effective public speaking skills. DOK2 1.5

- a Demonstrate effective communication in groups. 1.5.A
- b Demonstrate presentation skills. 1.5.B

6 Explore career opportunities within computer science in programming, cybersecurity, data science, robotics, artificial intelligence, human-computer interaction, and web development. 1.6

- a Identify and research career opportunities in programming, cybersecurity, data science, robotics, AI, and human-computer interaction. 1.6.A
- b Describe how career fields use technology in their work. 1.6.B
- c Examine the requirements, skills, wages, education, and employment opportunities in computer science career areas. 1.6.C

7 Create and maintain a personal portfolio website to showcase work and projects 1.7

Unit 2: Human-Computer Interaction 2.2

1 Explain the difference between computers and computing. DOK1 2.1

- a Identify characteristics of hardware components and their applications. 2.1.A
- b Explain the four characteristics of a computer: input, output, processing, and storage. 2.1.B
- c Explain the differences between tasks that can and cannot be accomplished with a computer. 2.1.C

2 Evaluate how the internet works and tools/methods used to navigate it. DOK3 2.2

- a Use appropriate tools and methods to execute internet searches. 2.2.A
- b Evaluate the reliability of websites and AI responses. 2.2.B
- c Define and give examples of the Internet of Things (IoT). 2.2.C

3 Analyze the effects of computing on society within economic, social, and cultural contexts. 2.3

- a Discuss legal, ethical, and security concerns raised by computing innovation. 2.3.A
- b Explain the implications of communication as data exchange. 2.3.B
 - 1 Recognize various forms of communication as data exchange. 2.3.B.1
 - 2 Describe the implications of data exchange on social interactions. 2.3.B.2
 - 3 Explain how computers are used for communications. 2.3.B.3
 - 4 Compare and contrast privacy and access concerns between online versus in-person data exchanges. 2.3.B.4
- c Identify web applications that influence society and education. 2.3.C
- d Identify appropriate and inappropriate use of social websites. 2.3.D

4 Explain the basic concepts of cloud computing and its applications. DOK2 2.4

- a Define “the cloud” and its major attributes. 2.4.A
 - b Describe the major benefits of cloud computing. 2.4.B
-

Unit 3: Problem Solving 3.3

1 Understand the problem-solving process. DOK2 3.1

- a Name and explain the steps in the problem-solving process. 3.1.A
 - b Solve various problems using the problem-solving process and document each step. 3.1.B
-

2 Design and interpret algorithms. DOK3 3.2

- a Define an algorithm and determine its effectiveness. 3.2.A
 - b Determine if a given algorithm successfully solves a stated problem. 3.2.B
 - c Create algorithms that meet specified objectives. 3.2.C
 - d Summarize the behavior of an algorithm. 3.2.D
 - e Compare the tradeoffs between different algorithms for solving the same problem. 3.2.E
 - f Explain the characteristics of problems that an algorithm cannot solve. 3.2.F
-

3 Demonstrate an understanding of binary numbers. DOK2 3.3

- a Explain the connections between binary numbers and computers. 3.3.A
 - b Count forward and backward in binary. 3.3.B
 - c Use binary digits to code and decode messages. 3.3.C
-

4 Understand simple search algorithms. DOK2 3.4

- a Illustrate and explain linear and binary search algorithms. 3.4.A
 - b Explain conditions in which each search would be appropriate. 3.4.B
-

5 Explain sorting algorithms. DOK2 3.5

- a Define sorted and unsorted lists. 3.5.A
 - b Describe various sorting algorithms and compare them. 3.5.B
-

6 Describe minimum spanning trees. DOK3 3.6

- a Solve minimum spanning tree problems and provide real-world examples (e.g., power grids, gas lines). 3.6.A
 - b Explain how a minimum spanning tree relates to computer science networks. 3.6.B
-

Unit 4: Introduction to Programming 4.4

1 Use appropriate algorithms to solve a problem. DOK3 4.1

- a Write steps or flow diagrams to plan solutions to programming problems. 4.1.A
- b Write code that is properly sequenced to solve problems. 4.1.B

2 Using a text-based language (i.e., Python, C++), design, code, test, and execute a program corresponding to a set of specifications. DOK2 4.2

- a Describe an event-driven program. 4.2.A
- b Use industry constructs such as pseudocode and comments to draft a program. 4.2.B
- c Apply programming structures such as variables, conditionals, loops, and input/output to create a program. 4.2.C

3 Using a text-based language, locate and correct errors in a program. DOK3 4.3

- a Deconstruct programs into smaller components to isolate problems. 4.3.A
- b Identify and correct errors in a program written by a student and another by a peer (debug). 4.3.B
- c Use AI to create and debug programs. 4.3.C
- d Evaluate the reliability of AI in creating and debugging programs. 4.3.D
- e Discuss the pros, cons, and ethics of using AI in programming. 4.3.E
- f Evaluate a peer's program and provide constructive feedback on accuracy, efficiency, and readability. 4.3.F

4 Compare/contrast at least three programming languages. DOK3 4.4

- a Identify the best use of each language and their differences. 4.4.A
- b Compare the syntax of each language for the following structures: variables, conditionals, loops, and input/output. 4.4.B

5 Use abstraction to reduce complexity. 4.5

- a Use abstraction to reduce complexity. 4.5.A
- b Explain abstraction and provide examples in everyday life. 4.5.B

Unit 5: Data Science and Computing 5.5

1 Understand big data and its characteristics. DOK1 5.1

- a Understand the complexities of collecting, processing, and managing large data sets. 5.1.A
- b Identify real-world applications of big data in various fields. 5.1.B
- c Utilize tools to analyze large data sets. 5.1.C
- d Draw conclusions about the data set selected. 5.1.D
- e Discuss methods of collecting and validating data. 5.1.E
- f Collaborate with others to create artifacts (i.e. surveys). 5.1.F
- g Understand and discuss data bias. 5.1.G

2 Identify and discuss the considerations that must be made for a large data set to be useful. 5.2

- a Consider how various data types (numbers, text, dates, etc.) lend themselves to processing. 5.2.A
- b Explain how different representations of data can tell different stories. 5.2.B
- c Collaborate with others to create, manage, and maintain a large data set. 5.2.C

3 Understand the complexities of collecting, processing, and analyzing data sets. DOK2 5.3

- a Identify the specific variables needed to analyze the data. 5.3.A
- b Interpret data and draw conclusions to solve problems. 5.3.B
- c Understand the problem-solving process. 5.3.C

4 Define data analytics. DOK1 5.4

- a Discuss what data analytics might involve. 5.4.A
- b Compare different analysis techniques and discuss the tradeoffs among them. 5.4.B
- c Understand machine learning and how it works. 5.4.C

Unit 6: Artificial Intelligence (AI) 6.6

1 Summarize artificial intelligence (AI) terms and concepts. DOK1 6.1

- a Explain key terminology associated with AI, including weak AI, strong AI, generative AI, artificial general intelligence (AGI), rule-based AI, and context-aware AI. 6.1.A
- b Develop an understanding of AI images and narratives. 6.1.B
- c Explore the concept of prompt engineering in AI. 6.1.C

2 Explore AI tools and their impact. DOK3 6.2

- a Identify the type of AI being used. (e.g., image recognition, speech recognition, translation, etc.) 6.2.A
- b Test various prompts and describe the results. 6.2.B
- c Understand how AI is changing different sectors. (e.g., medicine, agriculture, etc.) 6.2.C
- d Explore and explain the impact of AI on our society. 6.2.D
- e Recognize that future work is changing. 6.2.E

3 Recognize and understand AI data and bias. DOK3 6.3

- a Judge algorithmic bias and the effect of bias on individuals and society. 6.3.A
- b Examine issues involving privacy and the collection of data. 6.3.B

4 Develop skills in prompt engineering for AI applications. DOK2 6.4

- a Understand the basics of prompt engineering and its importance in AI. 6.4.A
 - b Create effective prompts for AI models to achieve desired outcomes. 6.4.B
 - c Evaluate the effectiveness of different prompts and refine them for better results 6.4.C
-

Unit 7a: Physical Computing—Robotics 7A.7A

1 Identify the criteria that describe a robot and determine if something is a robot. 7A.1

- a Describe how the design of a robot’s body affects its behavior. 7A.1.A
 - b Identify the parts and features of a robot (motors, sensors, batteries, buttons/switches, etc.). 7A.1.B
-

2 Build, code, and test a robot that solves a stated problem. 7A.2

- a Navigate the programming environment to build and code a robot that performs specific tasks. 7A.2.A
 - b Implement Boolean operators, loops, conditionals, and waits in robot programming to control behavior. 7A.2.B
 - c Debug coding of the robot by testing, identifying, and fixing errors to ensure proper functionality. 7A.2.C
-

Unit 7b: Physical Computing—Microcontrollers 7B.7B

1 Identify common microcontroller terms. DOK1 7B.1

- a Communicate microcontroller terms using multiple formats (e.g., verbally, textually, graphically). 7B.1.A
- b Identify and label the components of a hands-on or simulation microcontroller from the list below: 7B.1.B
 - 1 Power sources 7B.1.B.1
 - 2 Inputs 7B.1.B.2
 - 3 Switches 7B.1.B.3
 - 4 Push buttons 7B.1.B.4
 - 5 Sensors 7B.1.B.5
 - 6 Joysticks and remotes 7B.1.B.6
 - 7 Outputs 7B.1.B.7
 - 8 Buzzers 7B.1.B.8
 - 9 LEDs 7B.1.B.9
 - 10 LCD modules 7B.1.B.10

2 Use programming to manipulate microcontroller inputs and outputs. DOK2 7B.2

a Incorporate the following methods/concepts in the programming: 7B.2.A

1 Different languages (e.g., Scratch, SNAP, Python, etc.) 7B.2.A.1

2 Logic statements (e.g., if, and, or, not, etc.) 7B.2.A.2

3 Loops (e.g., for, if, while, etc.) 7B.2.A.3

3 Use a microcontroller for a specified purpose. DOK2 7B.3

a Demonstrate the proper use of a microcontroller for a specified purpose. 7B.3.A

b Explain how microcontrollers are used to manipulate a robotic system. 7B.3.B

4 Add microcontrollers and troubleshooting. DOK3 7B.4

a Describe the use of microcontrollers in physical computing. 7B.4.A

Unit 7c: Physical Computing—E-Textiles 7C.7C

1 Design, interpret, and evaluate simple circuit diagrams with key components. DOK2 7C.1

a Draw a circuit diagram with a simple circuit. 7C.1.A

b Identify necessary components of a simple circuit system (power, components, polarity, etc.) 7C.1.B

c Peer review one another's circuit diagrams according to a rubric. 7C.1.C

d Create a paper greeting card with electronic components (including designing it, crafting it, debugging it, and sharing it with classmates). 7C.1.D

2 Construct parallel circuits using conductive materials and switches while applying an iterative design process. 7C.2

a Use conductive thread to sew electronic components. 7C.2.A

b Design and create a working parallel circuit with three lights. 7C.2.B

c Demonstrate how a switch works to turn electricity flow on and off. 7C.2.C

d Use an iterative design process. 7C.2.D

3 Implement computational circuits with pre-programmed mini-computers and understand common grounding. DOK3 7C.3

a Design a computational circuit using a pre-programmed mini-computer. 7C.3.A

b Understand the role of a common ground. 7C.3.B

Unit 7d: Physical Computing—Networking 7D.7D

- 1 Explore fundamental networking concepts. DOK2** 7D.1
 - a Define what a computer network is and explain its purpose. 7D.1.A
 - b Identify different types of networks (LAN, WAN, WLAN) and their characteristics. 7D.1.B
 - c Explain the concept of IP addresses and their role in networking. 7D.1.C
 - d Describe the basic functions of networking devices such as servers, routers, and switches. 7D.1.D
 - e Demonstrate the ability to set up a simple local network and connect devices on paper, in a virtual setting, or with actual hardware. 7D.1.E
 - f Understand basic network security principles and identify common threats. 7D.1.F
-

Unit 7e: Physical Computing—VR System 7E.7E

- 1 Investigate the process of developing virtual environments. DOK2** 7E.1
 - a Create programs using virtual engines such as Unreal (visual) or Unity (text). 7E.1.A
 - b Demonstrate knowledge of key terms such as levels, worlds, scenes, assets, viewport, camera, etc. 7E.1.B
 - c Import assets for use in programming projects. 7E.1.C
 - d Script assets to perform designated tasks within a project. 7E.1.D
 - e Compile and publish completed projects for testing or as finished products. 7E.1.E
-

Unit 8: Cybersecurity 8.8

- 1 Define cybersecurity and its importance. DOK 1** 8.1
 - a Differentiate between attackers and defenders. 8.1.A
 - b Describe types of hacking and the CIA triad (Confidentiality, Integrity, Availability). 8.1.B
 - c Discuss ethics in cybersecurity. 8.1.C

- 2 Discuss confidentiality in cybersecurity.** 8.2
 - a Explain encryption, passwords, and multifactor authentication. 8.2.A
 - b Analyze case studies of data breaches. 8.2.B

- 3 Explain integrity and hashing. DOK2** 8.3
 - a Describe hashing and its role in maintaining data integrity. 8.3.A

- 4 Discuss availability considerations. DOK3** 8.4
 - a Explain backups, DoS (denial-of-service) attacks, disaster plans, and redundancy. 8.4.A

- 5 Analyze adversary thinking and social engineering. DOK4** 8.5
 - a Evaluate phishing, insider/outsider threats, and ransomware tactics. 8.5.A