Maryland CTE Programs of Study

Computer Programming I (2025)

GENERAL REQUIREMENTS. This course is recommended for students in Grades 9-12. 1

1 GENERAL REQUIREMENTS. This course is recommended for students in Grades 9-12. 1

INTRODUCTION 2

A Career and Technical Education (CTE) instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions. 2.A

B The Digital Technology Career Cluster focuses on building linkages in IT occupations for entry level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services. 2.B

C The Computer Programming Program of Study provides high school students with comprehensive software development skills through mastery of Python and Java programming languages. Students' progress from fundamental coding concepts to complex development projects while earning industry certifications and building professional portfolios. Graduates emerge prepared for both college computer science programs and immediate entry into software development careers, equipped with practical experience in current programming practices and software engineering principles. 2.C

D Computer Programming I introduces students to Python programming through hands-on development of algorithms and programs. Students master fundamental concepts like variables, data structures, and control flow while developing essential problem-solving and debugging skills. The course emphasizes professional practices including code documentation and collaboration, culminating in Python certification preparation. This foundation prepares students for advanced programming courses and establishes core competencies needed in software development. 2.D

E Students will participate in at least two Career-Connected Education and Work-Based Learning experiences in this course, which might include informational interviews or job shadowing relevant to the program of study. 2.E

Skip to main content

F **Students are encouraged to participate in extended learning experiences through aligned Career and Technical Student Organizations (CTSOs). CTSOs are a cocurricular requirement in the Carl D. Perkins Act, and alignment to CTSO activities is an expectation for CTE programs in the state of Maryland.** 2.F

KNOWLEDGE AND SKILLS 3

A **The student demonstrates the necessary skills for career development, maintenance of employability, and successful completion of course outcomes. The student is expected to:** 3.A

1. Identify and demonstrate positive work behaviors that enhance employability and job advancement, such as regular attendance, promptness, proper attire, maintenance of a clean and safe work environment, and pride in work. 3.A.1
2. Demonstrate positive personal qualities such as flexibility, open-mindedness, initiative, active listening, and a willingness to learn. 3.A.2
3. Employ effective reading, writing, and technical documentation skills. 3.A.3
4. Solve problems using critical thinking techniques and structured troubleshooting methodologies. 3.A.4
5. Demonstrate leadership skills and collaborate effectively as a team member. 3.A.5
6. Implement safety procedures, including proper handling of hardware and following cybersecurity guidelines. 3.A.6
7. Exhibit an understanding of legal and ethical responsibilities in the IT field, following data privacy laws and best practices for security. 3.A.7
8. Demonstrate time-management skills and the ability to prioritize tasks in a technical setting. 3.A.8

B **The student identifies various career pathways in the Computer Programming field. The student is expected to:** 3.B

1. Develop a career plan that includes the necessary education, certifications, job skills, and experience for specific roles in Computer Programming. 3.B.1
2. Create a professional resume and portfolio that reflect skills, projects, certifications, and recommendations. 3.B.2
3. Demonstrate effective interview skills for roles in IT and Computer Programming. 3.B.3

C **The student develops technology and digital literacy skills. The student is expected to:** 3.C

1. Use technology as a tool for research, organization, communication, and problem-solving. 3.C.1
2. Use digital tools, including computers, mobile devices, collaboration platforms, and cloud services, to access, manage, and create information. 3.C.2
3. Demonstrate proficiency in using emerging and industry-standard technologies, including virtualization tools, network management software, and cybersecurity applications. 3.C.3

Skip to main content

4 Understand ethical and legal considerations for technology use, including the principles of data protection, copyright, and responsible technology use. 3.C.4

D The student integrates core academic skills into computer programming practices. The student is expected to: 3.D

1 Demonstrate the use of clear communication techniques, both written and verbal, that are consistent with industry standards. 3.D.1

2 Apply fundamental mathematical operations and expressions, including basic arithmetic, variables, constants, and order of operations. Analyze patterns and relationships in data structures. Develop logical thinking through Boolean operations and flow control. 3.D.2

3 Investigate systematic problem-solving approaches through algorithmic thinking and debugging. Apply iterative testing methods to validate solutions. 3.D.3

E The student can explain basic computing concepts. The student is expected to: 3.E

1 Identify how computers represent information using binary. 3.E.1

2 Describe how computers follow instructions in programs. 3.E.2

3 Explain what an algorithm is and give everyday examples. 3.E.3

4 Discuss how the internet connects people and enables sharing of ideas. 3.E.4

5 Recognize how technology impacts daily life and future careers. 3.E.5

F The student can discuss how computing affects society. The student is expected to: 3.F

1 Identify positive ways technology is used in school, home, and community. 3.F.1

2 Recognize potential risks of sharing personal information online. 3.F.2

3 Discuss how apps and websites are designed to keep users engaged. 3.F.3

4 Identify technology access differences among different communities. 3.F.4

5 Describe how innovations in computing create new career opportunities. 3.F.5

G The student can use Python programming tools. The student is expected to: 3.G

1 Set up and open the Python programming environment. 3.G.1

2 Write and run simple Python statements. 3.G.2

3 Recognize when a program has errors and identify the error messages. 3.G.3

4 Add comments to explain what a program does. 3.G.4

5 Save and organize Python files for a project. 3.G.5

H The student can work with basic Python elements. The student is expected to: 3.H

1 Create variables to store information in a program. 3.H.1

2 Use different data types for different kinds of information (numbers, text, true/false). 3.H.2

3 Convert data from one type to another when needed. 3.H.3

Skip to main content

4 Perform basic math operations in Python programs. 3.H.4

5 Create and display messages to users. 3.H.5

I The student can create interactive programs. The student is expected to: 3.I

1 Get information from users with the input() function. 3.I.1

2 Create clear prompts that explain what input is needed. 3.I.2

3 Format output to make information easy to read. 3.I.3

4 Convert user input to the correct data type for processing. 3.I.4

5 Create a complete program that gets input, processes it, and shows results. 3.I.5

J The student can create programs that make decisions. The student is expected to: 3.J

1 Use if statements to run code only when certain conditions are true. 3.J.1

2 Add else statements to provide alternative actions. 3.J.2

3 Create programs with multiple conditions using elif. 3.J.3

4 Combine conditions using and, or, and not. 3.J.4

5 Create programs that respond differently based on user input. 3.J.5

K The student can create programs that repeat actions. The student is expected to: 3.K

1 Use for loops to repeat actions a specific number of times. 3.K.1

2 Create while loops that continue until a condition changes. 3.K.2

3 Use loops to process lists of information. 3.K.3

4 Avoid common loop problems like infinite loops. 3.K.4

5 Create programs that collect and process multiple pieces of information. 3.K.5

L The student can create and use lists. The student is expected to: 3.L

1 Create lists to store multiple pieces of related information. 3.L.1

2 Access specific items in a list using their position (index). 3.L.2

3 Add, remove, and change items in a list. 3.L.3

4 Find the highest, lowest, or average value in a list of numbers. 3.L.4

5 Sort lists to organize information. 3.L.5

Skip to main content

M **The student can process data in lists. The student is expected to:** 3.M

1. Use loops to work with each item in a list. 3.M.1
2. Use lists to keep track of information throughout a program. 3.M.2
3. Create simple programs that analyze data in lists. 3.M.3
4. Search for specific information in a list. 3.M.4
5. Create a program that solves a real problem using lists. 3.M.5

N **The student can create and use functions. The student is expected to:** 3.N

1. Break down a complex problem into smaller tasks that can be solved with functions. 3.N.1
2. Create functions that perform specific tasks when called. 3.N.2
3. Pass information to functions using parameters. 3.N.3
4. Return results from functions. 3.N.4
5. Document what a function does and how to use it. 3.N.5

O **The student can use Python libraries. The student is expected to:** 3.O

1. Import and use built-in Python modules. 3.O.1
2. Use the random module to create games or simulations. 3.O.2
3. Apply the math module to solve mathematical problems. 3.O.3
4. Use the datetime module to work with dates and times. 3.O.4
5. Identify when to use existing modules instead of writing new code. 3.O.5

P **The student can read and write files. The student is expected to:** 3.P

1. Open and read information from text files. 3.P.1
2. Write information to new text files. 3.P.2
3. Add information to existing files. 3.P.3
4. Process data from CSV files (like spreadsheet data). 3.P.4
5. Create programs that read, process, and write data to files. 3.P.5

Q **The student can make programs more reliable. The student is expected to:** 3.Q

1. Identify common types of errors in programs. 3.Q.1
2. Use try-except blocks to handle errors gracefully. 3.Q.2
3. Provide helpful error messages to users. 3.Q.3
4. Test programs with different kinds of input. 3.Q.4
5. Create programs that can recover from common errors. 3.Q.5

Skip to main content

The student can create complete Python programs. The student is expected to: 3.R

1. Plan a program that solves a specific problem. 3.R.1
2. Use appropriate Python features to implement a solution. 3.R.2
3. Test the program to make sure it works correctly. 3.R.3
4. Explain how their program works and why they made certain choices. 3.R.4
5. Create a final project that demonstrates their Python programming skills. 3.R.5