Maryland CTE Programs of Study

Computer Programming II - Java (2025)

GENERAL REQUIREMENTS. This course is recommended for students in Grades 10-12. 1

1 GENERAL REQUIREMENTS. This course is recommended for students in Grades 10-12. 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 II advances students' expertise through Java programming and object-oriented development principles. Students apply inheritance, polymorphism, and data structures to create complex applications while mastering professional debugging and testing strategies. The course emphasizes software engineering practices and collaborative development, culminating in Java certification preparation. This comprehensive foundation prepares students for both college computer science programs and industry roles. 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 Evaluate algebraic functions and equations, implement Boolean algebra and logic, apply basic statistics for data analysis, utilize different number systems (binary, decimal), calculate percentages and proportions, and employ basic geometry for graphics problems. 3.D.2

3 Apply systematic investigation and controlled testing methodologies. Analyze data collection methods and pattern recognition. Implement documentation processes for replicable results. 3.D.3

E The student can analyze Java language fundamentals. The student is expected to: 3.E

1 Compare and contrast primitive and reference data types in program implementation. 3.E.1

2 Construct programs using appropriate variable declarations and initialization. 3.E.2

3 Apply proper syntax for arithmetic operators and mathematical computations. 3.E.3

4 Evaluate type conversion requirements and implement casting. 3.E.4

5 Create programs utilizing the Scanner class for user input. 3.E.5

6 Design programs that format and display output effectively. 3.E.6

F The student can create object-oriented programming solutions. The student is expected to: 3.F

1 Design classes that model real-world objects with appropriate attributes and methods. 3.F.1

2 Implement multiple constructors for flexible object creation. 3.F.2

3 Apply encapsulation principles using public and private access modifiers. 3.F.3

4 Develop methods that effectively manipulate object state. 3.F.4

5 Create toString methods that properly represent object state. 3.F.5

6 Construct equals methods for object comparison 3.F.6

G The student can apply program control structures. The student is expected to: 3.G

1 Implement selection statements to control program flow. 3.G.1

2 Design iteration solutions using appropriate loop structures. 3.G.2

3 Create programs that process one-dimensional arrays. 3.G.3

Skip to main content

4 Develop algorithms for searching and sorting arrays. 3.G.4

5 Construct nested loop structures for complex iterations. 3.G.5

6 Generate solutions using two-dimensional arrays. 3.G.6

H The student can evaluate inheritance relationships. The student is expected to: 3.H

1 Design class hierarchies that model real-world relationships. 3.H.1

2 Create abstract classes and methods for code reuse. 3.H.2

3 Implement interfaces to define class behavior. 3.H.3

4 Apply polymorphic principles in program design. 3.H.4

5 Analyze when to use inheritance versus composition. 3.H.5

6 Develop programs utilizing method overriding. 3.H.6

I The student can develop data structure implementations. The student is expected to: 3.I

1 Create and manipulate ArrayList objects. 3.I.1

2 Implement common collection methods. 3.I.2

3 Compare efficiency of different data structures. 3.I.3

4 Design programs using appropriate data structures. 3.I.4

5 Apply iteration techniques for data processing. 3.I.5

6 Implement sorting and searching algorithms. 3.I.6

J The student can implement exception handling. The student is expected to: 3.J

1 Design robust error handling mechanisms. 3.J.1

2 Create custom exception classes. 3.J.2

3 Apply try-catch blocks appropriately. 3.J.3

4 Develop input validation routines. 3.J.4

5 Handle file I/O exceptions effectively. 3.J.5

6 Implement proper exception hierarchies. 3.J.6

K The student can synthesize file operations. The student is expected to: 3.K

1 Create programs that read from text files. 3.K.1

2 Develop programs that write to text files. 3.K.2

3 Implement file processing algorithms. 3.K.3

4 Design data persistence solutions. 3.K.4

5 Apply proper file closing procedures. 3.K.5

6 Handle file-related exceptions. 3.K.6

Skip to main content

L The student can evaluate program correctness. The student is expected to: 3.L

1 Design comprehensive test cases. 3.L.1

2 Implement debugging strategies. 3.L.2

3 Apply code review techniques. 3.L.3

4 Analyze program efficiency. 3.L.4

5 Create program documentation. 3.L.5

6 Validate program output. 3.L.6

M The student can create professional-quality code. The student is expected to: 3.M

1 Apply Java naming conventions. 3.M.1

2 Implement proper code organization. 3.M.2

3 Create clear documentation using Javadoc. 3.M.3

4 Design reusable code components. 3.M.4

5 Follow object-oriented design principles. 3.M.5

6 Apply best practices for code maintenance. 3.M.6