

CTE Standards Unpacking
Computer Programming I & II

Course: Computer Programming I & II

Course Description: Computer Programming I introduces students to the fundamentals of computer programming. Students will learn to design, code, and test their own programs while applying mathematical concepts. Teachers introduce concepts and problem solving skills through a programming language such as C, C++, C#, Java, Python, or Visual Basic. Computer Programming II reviews and builds on the concepts introduced in Computer Programming I and introduces students to more complex data structures. Topics include sequential files, arrays, and classes.

Career Cluster: Information Technology

Prerequisites: Computer Applications, Introduction to Information Technology Careers (recommended), Computer Hardware & Software (recommended)

Program of Study Application: Computer Programming is required for the Programming Pathway and recommended for the Networking & Hardware Pathway.

INDICATOR #CP 1: Identify and use a programming environment.		
SUB-INDICATOR 1.1 (Webb Level: 1): Demonstrate knowledge of external and internal computer hardware.		
SUB-INDICATOR 1.2 (Webb Level: 1): Demonstrate knowledge of software concepts.		
SUB-INDICATOR 1.3 (Webb Level: 2): Demonstrate the ability to compile, debug, and execute programs.		
Knowledge (Factual): <ul style="list-style-type: none"> Use the programming environment of a computer using hardware and software to write and execute programs. 	Understand (Conceptual): <ul style="list-style-type: none"> Understand the interrelatedness of the computer's internal and external hardware as well as its software to execute programs. 	Skills (Application): <ul style="list-style-type: none"> Use a programming language to compile, debug and execute a computer program.
Benchmarks <i>Students will be assessed on their ability to:</i> <ul style="list-style-type: none"> Diagram functions of the external and internal computer hardware. Differentiate between computer hardware and software. Compare and contrast software categories such as application software, web-based software, or operating system. Describe the difference between an interpreted language vs a compiled language. 		

- Demonstrate how to use the IDE to compile and run programs.
- Create a flowchart or a pseudocode for a problem to solve.
- Demonstrate how to debug programs.

Academic Connections

ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):

Language Arts Standards:

RI4. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text

RI7. Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.

W1. Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

W2. Write informative/explanatory texts to examine and convey complex ideas, concepts, and information clearly and accurately through the effective selection, organization, and analysis of content.

W4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Sample Performance Task Aligned to the Academic Standard(s):

*Students will explain orally or in a writing /multimedia presentation the differences between computer hardware and software.

Students will explain orally or in writing the various types of software categories.

*Students will explain orally or in a writing /multimedia presentation, the difference between an interpreted language and a compiled language.

W6. Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.

W8. Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation

W9. Draw evidence from informational texts to support analysis, reflection, and research.

SL2. Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data

SL4. Present information, findings, and supporting evidence, conveying a clear and distinct perspective, such that listeners can follow the line of reasoning, alternative or opposing perspectives are addressed, and the organization, development, substance, and style are appropriate to purpose and audience.

<p>INDICATOR #CP 2: Employ standard conventions for creation and design of a software program.</p>		
<p>SUB-INDICATOR 2.1 (Webb Level: 2): Demonstrate the ability to use a standard programming style.</p>		
<p>SUB-INDICATOR 2.2 (Webb Level: 2): Recognize software development processes.</p>		
<p>SUB-INDICATOR 2.3 (Webb Level: 1): Identify the syntactical components of a program.</p>		
<p>Knowledge (Factual):</p> <ul style="list-style-type: none"> • Components used to create and design a software programs using standard conventions. 	<p>Understand (Conceptual):</p> <ul style="list-style-type: none"> • Software programs use standard conventions. 	<p>Skills (Application):</p> <ul style="list-style-type: none"> • Create a software program.
<p>Benchmarks <i>Students will be assessed on their ability to:</i></p> <ul style="list-style-type: none"> • Employ syntax naming conventions. • Construct identifiers with meaningful formats. • Identify specifications and requirements used in creating a software program. • Decompose a problem into appropriate components. • Design solutions using algorithms and other problem solving techniques. • Identify syntactical components of a program. 		
<p>Academic Connections</p>		
<p>ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):</p> <p>Language Arts Standards:</p> <p>RI1. Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.</p> <p>RI4. Determine the meaning of words and phrases as they are used in a text,</p>	<p>Sample Performance Task Aligned to the Academic Standard(s):</p> <p>*In a writing/multimedia presentation, students will be able to identify specifications and requirements used in creating a software program.</p>	

<p>including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text</p> <p>RI7. Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.</p>	
--	--

<p>INDICATOR #CP 3: Properly use language-fundamental commands and operations.</p>		
<p>SUB-INDICATOR 3.1 (Webb Level: 2): Demonstrate the ability to use basic elements of a specific language.</p>		
<p>SUB-INDICATOR 3.2 (Webb Level: 2): Employ basic arithmetic expressions in programs.</p>		
<p>SUB-INDICATOR 3.3 (Webb Level: 3): Demonstrate the ability to use data types in programs.</p>		
<p>SUB-INDICATOR 3.4 (Webb Level: 2): Incorporate functions/methods.</p>		
<p>Knowledge (Factual):</p> <ul style="list-style-type: none"> Apply language fundamental commands and operations of a specific language to employ arithmetic expressions, data types, and functions and methods. 	<p>Understand (Conceptual):</p> <ul style="list-style-type: none"> Basic elements of a programming language. 	<p>Skills (Application):</p> <ul style="list-style-type: none"> Use programming language commands properly.
<p>Benchmarks <i>Students will be assessed on their ability to:</i></p> <ul style="list-style-type: none"> Demonstrate the ability to use input and output commands. Demonstrate the ability to use strings in programs. Use order of operation of expressions using basic arithmetic operators. 		

- Declare and use variables, constants, and enumerators.
- Differentiate between data types and their application.

Academic Connections

ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):

Language Arts Standards:

W2. Write informative/explanatory texts to examine and convey complex ideas, concepts, and information clearly and accurately through the effective selection, organization, and analysis of content.

W4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

W6. Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.

Sample Performance Task Aligned to the Academic Standard(s):

*In a presentation/multimedia presentation, Students will write basic function codes to show understanding of programming language.

* Students will use function codes and programming language to create computer programs.

INDICATOR #CP 4: Apply control structures.

SUB-INDICATOR 4.1 (Webb Level: 2): Demonstrate the ability to use relational and logical operators in programs.

SUB-INDICATOR 4.2 (Webb Level: 3): Investigate conditional statements.

SUB-INDICATOR 4.3 (Webb Level: 3): Implement loops in programs.

Knowledge (Factual):

- Use relational and logical operators, investigate conditional statements and implement loops into programs.

Understand (Conceptual):

- Advanced structures in programming languages.

Skills (Application):

- Incorporate and apply advanced control structures within a programming language.

<p>Benchmarks</p> <p><i>Students will be assessed on their ability to:</i></p> <ul style="list-style-type: none"> • Compare values using relational operators. • Form complex expressions using logical operators. • Incorporate If/Else structures and multiple way selections. • Use initial, terminal and incremental values in loops. • Implement nested and infinite loops.
--

Academic Connections

ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):	Sample Performance Task Aligned to the Academic Standard(s):
<p>Language Arts Standards:</p> <p>W2. Write informative/explanatory texts to examine and convey complex ideas, concepts, and information clearly and accurately through the effective selection, organization, and analysis of content.</p> <p>W4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.</p> <p>W6. Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.</p>	<p>*Students will write a code using initial, terminal, and incremental values in loops.</p> <p>*In a presentation/multimedia presentation, Students will explain how to incorporate and apply advanced control structures within a programming language.</p>

<p>INDICATOR #CP 5: Explore career opportunities in programming.</p>
<p>SUB-INDICATOR 5.1 (Webb Level: 1): Identify personal interests and abilities related to Computer Programming/Software Engineering careers.</p>
<p>SUB-INDICATOR 5.2 (Webb Level: 3): Investigate career opportunities, trends, and requirements related to computer programming/software engineering careers.</p>
<p>SUB-INDICATOR 5.3 (Webb Level: 2): Demonstrate job skills for programming industries.</p>

<p>Knowledge (Factual):</p> <ul style="list-style-type: none"> Skills, interests, and abilities related to Computer Programming/Software Engineering careers. 	<p>Understand (Conceptual):</p> <ul style="list-style-type: none"> Career options available in Computer Programming/Software Engineering. 	<p>Skills (Application):</p> <ul style="list-style-type: none"> Explore how their skills, interests, and abilities match Computer Programming/Software Engineering careers.
---	---	---

Benchmarks

Students will be assessed on their ability to:

- SDMyLife assessments including career matchmaker and ability profiler.
- Consider the financial impact of a Computer Programming/Software Engineering career.
- Display ability to work as part of a team and take direction from others.

Academic Connections

<p>ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):</p> <p>Language Arts Standards:</p> <p>RI1. Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.</p> <p>RI4. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text</p> <p>RI7. Integrate and evaluate multiple sources of information presented in different</p>	<p>Sample Performance Task Aligned to the Academic Standard(s):</p> <p>*Students will orally or in a writing/multimedia project explain the career of a Computer Programmer/Software Engineer.</p> <p>*In reflective writing, assignment, students will assess the qualities of a Computer Programmer/Software Engineer. Students will reflect on their own personal knowledge to assess if the career path is right for them, or what they need to accomplish to make this their career path.</p>
--	---

media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.

W1. Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

W2. Write informative/explanatory texts to examine and convey complex ideas, concepts, and information clearly and accurately through the effective selection, organization, and analysis of content.

W4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

W6. Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.

W8. Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation

W9. Draw evidence from informational texts to support analysis, reflection, and

<p>research.</p> <p>SL2. Integrate multiple sources of information presented in diverse formats and media (e.g., visually, quantitatively, orally) in order to make informed decisions and solve problems, evaluating the credibility and accuracy of each source and noting any discrepancies among the data</p> <p>SL4. Present information, findings, and supporting evidence, conveying a clear and distinct perspective, such that listeners can follow the line of reasoning, alternative or opposing perspectives are addressed, and the organization, development, substance, and style are appropriate to purpose and audience.</p>	
--	--

INDICATOR #CP 6: Integrate arrays. (Computer Programming II)		
SUB-INDICATOR 6.1 (Webb Level: 2): Demonstrate the ability to use arrays in programs.		
SUB-INDICATOR 6.2 (Webb Level: 3): Demonstrate the ability to use strings in programs.		
Knowledge (Factual): <ul style="list-style-type: none"> Use standard advanced conventions to create and design a software program integrating arrays. 	Understand (Conceptual): <ul style="list-style-type: none"> Advanced software programs use standard conventions and integrated arrays. 	Skills (Application): <ul style="list-style-type: none"> Create an advanced software program integrating arrays.
Benchmarks <i>Students will be assessed on their ability to:</i> <ul style="list-style-type: none"> Declare and initialize arrays. Add and remove items from arrays. 		

- Compare and concatenate string identifiers.
- Locate substring positions.

Academic Connections

ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):	Sample Performance Task Aligned to the Academic Standard(s):
<p>Language Arts Standards:</p> <p>RI1. Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.</p> <p>RI4. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text</p> <p>RI7. Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.</p>	<p>*After consulting several sources, students will compare the various types of string identifiers and assess their strengths and weaknesses</p> <p>*Students will create their own advanced software program that includes standard and advanced conventions and integrating arrays.</p>

<p>INDICATOR #CP 7: Implement object-oriented programming techniques. (Computer Programming II)</p>
<p>SUB-INDICATOR 7.1 (Webb Level: 3): Demonstrate the ability to use existing classes.</p>
<p>SUB-INDICATOR 7.2 (Webb Level: 4): Demonstrate the ability to create user-defined classes.</p>
<p>SUB-INDICATOR 7.3 (Webb Level: 4): Demonstrate proper design principles with classes.</p>

<p>Knowledge (Factual):</p> <ul style="list-style-type: none"> • Create and implement existing and user-defined classes using proper design principles. 	<p>Understand (Conceptual):</p> <ul style="list-style-type: none"> • Object-oriented programming techniques. 	<p>Skills (Application):</p> <ul style="list-style-type: none"> • Create a software program implementing object-oriented programming techniques.
<p>Benchmarks <i>Students will be assessed on their ability to:</i></p> <ul style="list-style-type: none"> • Instantiate objects use object data and incorporate functions. • Create, initialize, and use data members. • Create classes using proper modifiers and accessors. 		
<p><i>Academic Connections</i></p>		
<p>ELA Literacy and/or Math Standard (if applicable, Science and/or Social Studies Standard):</p> <p>Language Arts Standards:</p> <p>RI1. Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.</p> <p>RI4. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text</p> <p>RI7. Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.</p>	<p>Sample Performance Task Aligned to the Academic Standard(s):</p> <p>*Students will consult and use a manual to create classes using proper modifiers and accessors.</p> <p>*Students will create a software program implementing object-oriented programming techniques.</p> <p>*Students will orally or in a writing/multimedia project explain how to implement object-oriented programming techniques.</p>	



Additional Resources

Please list any resources (e.g., websites, teaching guides, etc.) that would help teachers as they plan to teach these new standards.

<http://www.acm.org>

<http://www.ieee.org>

<http://marshallbrain.com/kids-programming.htm>