

COMPUTER SCIENCE
College Credit and Careers Network
Dual Credit Articulation Agreement

Upon completion of high school courses equivalent to the following competencies:

- ❑ Solve problems applying the techniques of decomposition (breaking a problem into smaller more manageable pieces), critical reasoning, induction and deduction.
- ❑ Define what an Algorithm is, and explain its five characteristics of finiteness, definiteness, effective, zero+ inputs and one+ output.
- ❑ Describe the functionality of memory, CPU and programs, including compilation, linking/loading and executing, and history of computing.
- ❑ Identify and explain the Software Development Life Cycle.
- ❑ Describe and apply the main steps of specification, analysis, design, implementation, testing and maintenance to solve problems.
- ❑ Identify the main goals of software programming, including correctness, maintainability, style, structure and documentation.
- ❑ Define the concepts of data types, including the fundamental data types (int, float, bool, and char) as well as compound data types (string, arrays, records and classes).
- ❑ Use the fundamental and compound data types to model data in problem solving and programs.
- ❑ Explain the concept of type conversions, and apply the knowledge of declaration and use of variables, constants, executable statements as well as input, output of values (including boolean values), to write a simple program.
- ❑ Explain the process of evaluation of arithmetic expressions in conjunction with the precedence rules for the operators.
- ❑ Explain the sequence, selection, iteration and recursion control structures and use these constructs in simple programs.
- ❑ Describe the process of procedural abstraction in terms of separation of design and purpose from implementation.
- ❑ Explain the black box thinking and top down design in designing the solutions for solving the problems.
- ❑ Explain how to define and invoke functions, and apply in simple programs.
- ❑ Describe the process of parameter passing, distinguishing between actual and formal parameters and the process of returning values.
- ❑ Explain the parameter passing mechanisms of call by value, call by address and call by reference and describe the scope of identifiers.
- ❑ Discuss the availability of functions using function libraries and reusability of code, and use library functions and existing code in programs.
- ❑ Describe the concept of logical and physical files and operations that can be performed on them, like read, write, open and close, and use these concepts in programs.
- ❑ Discuss the importance and role of documentation using comments and descriptive identifiers, and incorporate documentation techniques in programs.
- ❑ Explain the various techniques used in testing and debugging programs, like desk checking and test suites.
- ❑ Apply testing concepts and techniques to verify and fix errors in their own programs.
- ❑ Explain the importance of being able to read, extend and modify the existing code, and apply these principles in program development.
- ❑ Use the programming environment tools to create programs using an editor, compile or interpret them, and use the debugger to find and fix errors

A student earning a “B” or better may earn college credit at the following college:

<u>College</u>	<u>Course</u>	<u>Credits</u>
Bellevue College	CS 210	5