CS A150: C++ PROGRAMMING LANGUAGE 1

Item	Value
Curriculum Committee Approval Date	12/06/2023
Top Code	070710 - Computer Programming
Units	4 Total Units
Hours	90 Total Hours (Lecture Hours 63; Lab Hours 27)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	Yes
Basic Skills	Not Basic Skills (N)
Repeatable	No
Grading Policy	Standard Letter (S), • Pass/No Pass (B)

Course Description

First course in ANSI/ISO Standard C++ programming language. Topics include data types, strings, operators, expressions, control flow, input/output, functions, pointers, arrays, preprocessor, streams, enumerated data type, dynamic memory allocation, objects, classes, vectors, inheritance, object-oriented design and recursion. PREREQUISITE: CS A100 or CS A122 or CS A131 or CS A170. Transfer Credit: CSU; UC.

Course Level Student Learning Outcome(s)

1. Use C++ to solve problems that incorporate the use of correct data types, arrays, loops, branching, and operators.
2. Use C++ to solve problems that use classes and functions.

Course Objectives

- 1. Apply structured programming techniques to solve various problems using the C++ language.
- 2. Apply the correct data type to solve various problems.
- 3. Apply various forms of arrays, vectors and enumerated data type to solve problems.
- 4. Apply recursion and iteration to solve various problems.
- 5. Produce programs that use the concept of dynamic memory allocation.
- 6. Produce programs using various forms of loops and branching statements.
- 7. Produce programs that use various operators that exist in the C++ language.
- 8. Produce programs which show students understanding of how data is passed between various modules without resorting to global variables.
- 9. Produce programs which use sequential access file I/O concepts.
- 10. Apply object-oriented methodology to solve problems.
- 11. Apply inheritance and polymorphism to solve a variety of problems.

Lecture Content

Introduction to C++ Writing and compiling a simple C++ program Fundamental data types Numeric types – integer and floating-point Variables, assignment, arithmetic operators, input, output and comments Cast operator Combining assignment and arithmetic operators Const variables, enumerated data type Using functions String variables, string member functions Formatted output Using objects Use simple classes defined by the author to learn how to work with objects Constructors, calling member functions using the dot operator Use simple graphics classes defined by the author to draw simple shapes Basic control flow If statement, selector operator, relational operators Comparison of floating-point values Input validation While loop Correctness proofs Functions Function definition and function comments Function prototype Passing arguments by value and by reference Const reference parameters Assertion Classes Class definition, function implementation Constructors, mutator functions and accessor functions Constructors with field initializer list Member and non-member functions Separate compilation Advanced control flow Multiple alternatives and nested if statements Switch statement Boolean operators For-loop and do-while loop Nested loops Processing text input Redirection and pipes Simulations and approximations Testing and debugging Designing test harnesses for testing components of your program Debugger Vectors and arrays i >Using vectors to solve problems Passing and returning vectors to and from functions Using arrays to solve problems Two-dimensional arrays Pointers Dynamic memory allocation using the new operators Pointer variable definition, dereferencing operator and address-of operator The pointer Deallocating memory using the delete operator Inheritance Defining derived classes Derived class constructor calling base class constructor Protected access Polymorphism Virtual functions Streams Reading and writing text files String streams Processing command-line arguments Sequential vs random access Object-oriented design Software life cycle Concepts of cohesion and coupling CRC card method UML class diagrams Recursion Relationship between recursion and iteration How to “think recursively” Recursive helper functions i

Lab Content

Programming labs and exercises should include: Design, code and run a C++ input, processing and output program. Demonstrate the use of different data types and C++ operators to produce functions with parameters and return values. Write functions which use simple and sequential selection and logical operators. Write functions using the string member functions to find and manipulate substrings.

Method(s) of Instruction

- Lecture (02)
- DE Online Lecture (02X)
- Lab (04)
- DE Online Lab (04X)

Instructional Techniques

Lecture/discussion

Reading Assignments

Students will spend a minimum of 6 hours per week completing weekly programming assignments.

Writing Assignments

Students will spend a minimum of 6 hours per week writing code.

Out-of-class Assignments

Students will spend a minimum of 6 hours per week writing code.

Demonstration of Critical Thinking

Laboratory exercises.

Required Writing, Problem Solving, Skills Demonstration

Written examination

Eligible Disciplines

Computer science: Masters degree in computer science or computer engineering OR bachelors degree in either of the above AND masters degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelors degree in engineering AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in mathematics AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in any of the above AND a masters degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum qualification purposes, under the discipline of the application. Masters degree required.

Textbooks Resources

1. Required Horstmann, C.. Big C++, Latest ed. New Jersey: Wiley, 2020
2. Required Dale, N., Weems, C., Richards, T.. Programming and Problem Solving with C++, 7th ed. Jones Bartlett, 2023