Course Outline for Computer Science 14

INTRODUCTION TO STRUCTURED PROGRAMMING IN C++

Catalog Description:

CSCI 14 - Introduction to Structured Programming In C++  
4.00 units
Introduction to structured programming and problem solving using the C++ language. Problem solving techniques, algorithm design, testing and debugging techniques, and documentation standards. C++ syntax: elementary operators, data types, control structures, user-defined and library functions, basic input/output, sequential files, arrays and structs. Appropriate for students with little or no programming experience, but comfortable using computers with modern GUI operating systems.

Prerequisite: MTH 55 or , MTH 55B or , MTH 55L or , MTH 54 or , MTH 54L (completed with a grade of "C" or higher) or an appropriate skill level demonstrated through the Mathematics Assessment process

Strongly Recommended: CSCI 7 (completed with a grade of "C" or higher) or , ENGL 1A

Grading Option: Letter Grade

Discipline:

<table>
<thead>
<tr>
<th>Units</th>
<th>Contact Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Week</td>
</tr>
<tr>
<td>Lecture</td>
<td>3</td>
</tr>
<tr>
<td>Laboratory</td>
<td>3</td>
</tr>
<tr>
<td>Clinical</td>
<td>0.00</td>
</tr>
<tr>
<td>Total</td>
<td>4.00</td>
</tr>
</tbody>
</table>

Prerequisite Skills:
Before entry into this course, the student should be able to:
1. identify functions, find their domains and ranges, and use function notation in the context of real data;
2. perform operations using the properties of rational exponents;
3. solve exponential equations using logarithms;
4. graph quadratic, power, and logarithmic functions;
5. choose an appropriate model for a realistic situation given a choice of mathematical models.
6. perform operations using the properties of rational exponents;
7. graph exponential functions and interpret real growth and decay situations and data with exponential functions;
8. solve exponential equations using logarithms;
9. graph quadratic, power, and logarithmic functions;
10. choose an appropriate model for a realistic situation given a choice of mathematical models.
11. solve exponential and logarithmic equations;
12. apply the concepts of logarithmic and exponential functions;
13. sketch the graphs of functions and relations:
   a. algebraic, including polynomial and rational
   b. logarithmic
   c. exponential
   d. circles;
14. solve exponential and logarithmic equations;
15. apply the concepts of logarithmic and exponential functions;
16. apply the properties of and perform operations with radicals;
17. solve equations and inequalities involving absolute values;
18. solve equations involving radicals;
19. find the distance between two points;
20. solve equations in quadratic form;
21. state domain and range of a function or given the graph of a function

Measurable Objectives:
Upon completion of this course, the student should be able to:
1. when using the college computer laboratory, follow the procedures to sign in and out;
2. write, edit, compile, run and debug programs;
3. demonstrate steps involved in program development;
4. write simple C++ data types in programs and apply how they are represented in the machine;
5. write C++ expressions using selected operators, and apply the rules of precedence used in their evaluation;
6. apply the structured programming constructs: sequence, selection and iteration;
7. perform elementary interactive input and output operations;
8. code void and value-returning functions with value and reference parameters and use them in a program;
9. define and use the structured C++ data types: array, string, struct in applications drawn from mathematics, the sciences, and other areas;
10. use text files to record and retrieve information in elementary applications;
11. produce well-documented, user-friendly programs of short to medium length.

Course Content:
1. Course Content (Lecture):
   A. Review of program development
      a. Top-Down design
   B. Elementary Data types in C++
      a. Simple data types: int, unsigned, long, char, float, double
      b. Machine representation of the simple data types: int, long and char (Optional)
   C. Expressions and assignment statements in C++
      a. Selected C++ operators: Arithmetic, Logical, Assignment, Relational, size of
   D. Selection and iteration structures
      a. IF and IF-ELSE statements
      b. SWITCH statements
      c. Syntax of FOR, WHILE and DO WHILE loops
   E. Elementary interactive input and output in C++
      a. cin and cout
      b. I/O of numeric, char, string
      c. Using sentinel controlled loop to read from the keyboard
      d. Using an EOF controlled loop to read from a file
   F. Use of functions in modular programming
      a. Role of program modules in well designed programs
      b. Using built-in library functions
      c. Function declarations and calls
      d. Parameter passing mechanisms: value vs. reference
      e. Value returning vs. void functions
   G. One dimensional arrays
      a. Definition of array and motivation for use
      b. Input fixed or varying sized data set into an array from keyboard or file
      c. How to pass arrays as parameters
      d. Other array manipulation such as summing values
      e. C-String, an array of characters
      f. Binary search and elementary sorts
      g. Mention array names in relation to pointer variables (Optional)
   H. Structs (Required), Array of Structs (Optional)
      a. Syntax and Declarations
      b. How to define and access Structs
      c. Compares and contracts with arrays
      d. Struct as function parameters
   I. Sequential files
      a. Definition of text file
      b. How to define, open, and close text files
      c. Elementary reading/writing text files (loop/EOF)
      d. How to pass a file variable as a function parameter
   J. User-friendly programs
      a. Input Validation techniques
      b. Output prompts
      c. Output design

2. Course Content (Laboratory):
   A. Rules and procedures when using College Computer Laboratory
      a. Signing in and out procedures
   B. Use the C++ integrated development environment
      a. Create, edit, compile and run programs
      b. Provide program comments and documentation
      c. Test and debug programs
      d. Evaluate program efficiency

Methods of Presentation
1. Student use of appropriate computer laboratory
2. Lecture/Discussion
3. Demonstration/Exercise

Assignments and Methods of Evaluating Student Progress

1. Typical Assignments
   A. Revise your previous angle assignment so that the main program does little but perform tile initialization, loop control and call functions.
   B. This assignment uses material from chapter 8 on files. In this assignment folder you will find a slightly revised version of copyfile.cpp. This is figure 8.4 on p. 397 – p. 399. Revise this program so that each line contains a line number followed by a colon and a couple of spaces. The original text should still line up exactly as it did before.

2. Methods of Evaluating Student Progress
   A. Quizzes
   B. Midterm Examination
   C. Final Examination
   D. Assigned programs
   E. Homework

3. Student Learning Outcomes
   Upon the completion of this course, the student should be able to:
   A. Demonstrate steps involved in program development
   B. Produce well-documented, user-friendly programs of short to medium length
   C. Understand the mechanics under C++ of passing argument values by values and by reference to a function. Given the relevant code, the student will demonstrate their knowledge of parameter passing by correctly determining what would be output of that function.

Textbook (Typical):
Special Student Materials
1. Portable storage device such as a USB flash drive

Abbreviated Class Schedule Description:

Introduction to structured programming and problem solving using the C++ language. Problem solving techniques, algorithm design, testing and debugging techniques, and documentation standards. C++ syntax: elementary operators, data types, control structures, user-defined and library functions, basic input/output, sequential files, arrays and structs. Appropriate for students with little or no programming experience, but comfortable using computers with modern GUI operating systems.

Prerequisite: MTH 55 or MTH 55B or MTH 55L or MTH 54 or MTH 54L (completed with a grade of "C" or higher) or an appropriate skill level demonstrated through the Mathematics Assessment process

Strongly Recommended: CSCI 7 (completed with a grade of "C" or higher) or ENGL 1A