

Course Prefix/Number: CPT 232
Course Title: C++ Programming I
Lec Hours/Week: 3.0
Lab Hours/Week: 0.0
Credit Hours/Semester: 3.0

[Distance Learning Attendance/VA Statement](#)
[Textbook Information](#)

COURSE DESCRIPTION

This introductory course in C++ Programming I emphasizes the designing, coding, testing, and debugging of C++ programs involving input/output operations, data types, storage classes, decision structures, looping, functions, arrays, simple pointers, and strings.

COURSE COMPETENCIES

Upon successful completion of this course, the student should be competent to complete the following tasks:

Module 1 - Basic Operations and Variables

- Produce screen output using the cout object from iostream.h
- Obtain input from the keyboard using the cin object from iostream.h
- Describe the differences between the basic data types used in C++
- Declare and initialize variables
- Evaluate mathematical expressions and store the resulting value in a variable using arithmetic and assignment operators.
- Document code with well-placed, descriptive comments
- Find and fix syntax and logic errors in programs that produce input and output and use arithmetic operations

Module 2 - Control Structures

- Solve problems requiring the use of nested if/else statements
- Create logical expressions with relational and logical operators
- Solve problems requiring the use of while or do/while loops
- Solve problems using for loops
- Solve problems using the switch construct
- Find and fix syntax and logic errors in programs containing decisions and loops

Module 3 - Functions

- Given the prototype for a pre-defined function, write a function call
- Write functions using call-by-value and call-by-reference parameters
- Explain the relative benefits of using call-by-value and call-by-reference
- Relate the effect of functions on program design and testing
- Describe and effectively use variables in the auto, extern, and static storage classes
- Find and fix syntax and logic errors in programs which use pre-defined and user defined functions

Module 4 - Arrays

- Declare and initialize single-subscripted and multi-subscripted arrays
- Input, output and manipulate the values in an array
- Create functions that take arrays as parameters
- Describe how to search an array
- Find and fix syntax and logic errors in programs containing arrays

Module 5 - Pointers

- Declare and initialize pointers
- Describe the four ways to send pointers to functions using the const keyword
- Describe the relationship between pointers and arrays
- Manipulate pointers using pointer arithmetic
- Manipulate strings of characters using functions from cstring.h
- Find and fix syntax and logic errors in programs containing pointers

MINIMAL STANDARDS

Minimal standards of performance on all course competencies for receiving credit for the course and indicated by 60% overall accuracy on evaluation instruments that address the course competencies listed above. Required standards of performance on all course competencies for enrollment in subsequent higher-level computer technology courses are indicated by 70% overall accuracy on evaluation instruments that address the course competencies listed above.

COURSE REQUIREMENTS

Students are responsible for attending all schedule class meetings until they have completed all course requirements. Students are responsible for all material covered and for all assignments made in all classes. Any student caught cheating or involved in other academic dishonesty will be given a grade of zero and will be subject to further disciplinary action.

ATTENDANCE POLICY

The attendance policy as stated in the York Technical College Handbook will be enforced. Make-up tests will not be given for theory tests. If a student must miss a theory test, he/she will get a zero for that test. However, students have the option of taking the comprehensive final. The student's grade on the comprehensive final will replace his/her lowest theory test grade. It is the student's responsibility to schedule a time for a make-up hands-on test with his/her instructor.

EVALUATION STRATEGIES/GRADING

| Module 1 (20% total) | Module 2 (20% total) | Grading Scale | |
|---|--|---------------|---|
| | | 90-100 | A |
| Test – 10% Program(s) – 5% Homework – 5% | Test – 10% Program(s) – 5% Homework – 5% | 89-89 | B |
| Module 3 (20% total) | Module 4 (20% total) | 70-79 | C |
| | | 60-69 | D |
| Test – 10% Program(s) – 5% Homework – 5% | Test – 10% Program(s) – 5% Homework – 5% | Below 60 | F |
| Module 5 (20% total) | | | |
| Test – 10% *Program(s) – 5% Homework – 5% | | | |

*Completion of Module 5 Program(s) is required to receive a grade for the course

ENTRY LEVEL SKILLS

A student entering this course should be familiar with structured programming concepts, have adequate flowcharting skills and be familiar with the Windows environment.

PREREQUISITES: CPT 114 and CPT 168 with minimum grades of "C"

CO-REQUISITES: None

Disabilities Statement: Any student who feels s/he may need an accommodation based on the impact of a disability should contact the Special Resources Office (SRO) at 803-327-8007 in the 300 area of Student Services. The SRO coordinates reasonable accommodations for students with documented disabilities.

TOPIC/CONTENT OUTLINE

Module 1 Basic Operations and Variables

- A. Producing screen output using the cout object from iostream.h
- B. Obtaining input from the keyboard using the cin object from iostream.h
- C. Describing the differences between the basic data types used in C++
- D. Declaring and initialize variables
- E. Using arithmetic and assignment operators to evaluate mathematical expressions and store the resulting value in a variable
- F. Documenting code with well-placed, descriptive comments
- G. Finding and correcting syntax and logic errors in programs that produce input and output and use arithmetic operations

Module 2 Control Structures

- A. Solving problems that require the use of nested if/else statements
- B. Creating logical expressions with relational and logical operators
- C. Solving problems that require the use of while or do/while loops
- D. Using for loops to solve problems
- E. Using the switch construct to solve problems
- F. Finding and correcting syntax and logic errors in programs that contain decisions and loops

Module 3 Functions

- A. Writing function calls given the prototype for pre-defined functions
- B. Writing functions that use call-by-value and call-by-reference parameters
- C. Explaining the relative benefits of call-by-value and call-by-reference
- D. Relating the effect of functions on program design and testing
- E. Describing and effectively using variables in the auto, extern, and static storage classes
- F. Finding and correcting syntax and logic errors in programs that use predefined and user-defined functions

Module 4 Arrays

- A. Declaring and initializing single-subscripted and multi-subscripted arrays
- B. Obtaining input, producing output and manipulating array values
- C. Creating functions that take arrays as parameters
- D. Describing how to search arrays
- E. Finding and correcting syntax and logic errors in programs that contain arrays

Module 5 Pointers

- A. Declaring and initializing pointers
- B. Describing the four ways to send pointers to functions using the const keyword
- C. Describing the relationship between pointers and arrays
- D. Manipulating pointers using pointer arithmetic
- E. Manipulating strings of characters using functions from string.h
- F. Finding and correcting syntax and logic errors in programs that contain pointers