

**Course Prefix/Number:** CPT 233  
**Course Title:** C++ Programming II  
**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 course introduces object-oriented design techniques using C++. Topics include classes, friends, overloading operators, inheritance, and virtual functions.

## **COURSE COMPETENCIES**

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

### **Module 1 - Introduction to Classes**

- Describe the design benefits of user-defined classes
- Relate the differences between public and private class members
- Define and create constructors
- Define and create destructors
- Define and create accessor functions
- Define and create mutator functions
- Define and create friend functions
- Describe the usefulness of static class members
- Define the this pointer and use it to cascade function calls
- Define class composition and compose two or more classes

### **Module 2 -Operator Overloading**

- Define operator overloading
- Describe the benefits of overloading operators
- Overload operators with class members
- Overload operators with friends of the class

### **Module 3 - Inheritance**

- Define inheritance
- Describe the object-oriented design benefits gained through use of inheritance
- Create inheritance hierarchy charts for related classes
- Explain the difference between composition and inheritance
- Describe the difference between public, protected, and private class members
- Override base class members in a derived class
- Write classes with at least two levels of inheritance
- Describe the differences between public, protected, and private inheritance
- Cast derived class objects as base class objects

### **Module 4 - Virtual Functions**

- Define virtual functions
- Explain the object-oriented design benefits gained through use of virtual functions
- Create classes with virtual functions
- Create classes with pure virtual functions

## 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 scheduled 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 (25% total)  Tests – 12.5% of final average Program(s)/Homework – 12.5% of final average	Module 2 (25% total)  Test(s) – 12.5% of final average Program(s)/Homework – 12.5% of final average	Grading Scale	
		90-100	A
Module 3 (25% total)  Test(s) – 12.5% of final average Program(s)/Homework – 12.5% of final average	Module 4 (25% total)  Test(s) – 12.5% of final average *Program(s)/Homework – 12.5% of final average	89-89	B
		70-79	C
		60-69	D
		Below 60	F

\*Completion of Module 4 Program(s) is required to receive a grade for the course

## ENTRY LEVEL SKILLS

A student entering this course should be proficient with procedural programming in C++.

**PREREQUISITE:** CPT 232 – minimum grade 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.