

## **COURSE SYLLABUS**

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 are 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.

Students enrolled in a degree or certificate program in Computer Technology must attain 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. Students who are absent from a class more than 20% of the hours assigned will be withdrawn. 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.

#### **Evaluation Strategies/Grading**

A minimum of four tests and four programming assignments will be given. Tests will make up 50% of the student's final grade. The programs will make up 30% of the student's final grade. Homework assignments and quizzes will account for the remaining 20% of the student's grade.

#### **Grading Scale**

90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
Below 60	F

#### **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