NVCC COLLEGE-WIDE COURSE CONTENT SUMMARY
ITP 232
C++ Programming II (4 cr.)

Course Description

ITP 232 - Presents in-depth instruction of advanced object-oriented techniques for data structures using C++. Lecture 4 hours per week.

Recommended Co-requisites or Pre-requisites

ITP 132

Course Objectives

Upon completion of this course, the student will be able to:

- Create applications demonstrating overloaded operators and functions.
- Create applications demonstrating inheritance.
- Create applications demonstrating dynamic memory allocation using data structures such as linked lists and trees.

Course Content

- Encapsulation Concepts
- Method and Operator Overloading
- Inheritance and Abstract Classes
- Polymorphism and Virtual Functions
- Stream Input and Output
- Pointers and Dynamic Allocation
- Data Structures
- Exception Handling

Student Learning Outcomes

Encapsulation Concepts

- Understand the object-oriented concept of encapsulation
- Understand and code the components of a Class
- Describe the various access-specifiers of a class
- Define the scope and visibility of the class components

Method and Operator Overloading

- Describe the fundamentals of method overloading
- Explain how to code overloaded methods
- Understand the fundamentals of operator overloading
- Recognize the differences between operator methods as class members vs. as friend methods
- Correctly code overloaded stream-insertion and stream-extraction operators
- Software engineer overloaded unary and binary operators in C++
Inheritance and Abstract Classes

- Understand the object-oriented concept of inheritance
- Explain the differences between public, protected and private inheritance
- Explain the relationships between base and derived classes
- Explain the functionality of constructors and destructors in a derived class
- Software engineer inheritance applications in C++
- Explain the concept of abstract classes

Polymorphism and Virtual functions

- Understand the concept of polymorphism
- Explain and code virtual destructors
- Software engineer applications with polymorphism, virtual functions and dynamic binding

Stream Input and Output

- Explain the concept of streams in C++
- Understand the difference between classic and standard streams
- Describe input and output classes and objects
- Discuss the various stream manipulators and format states
- Software engineer applications utilizing streams

Pointers and Dynamic Allocation

- Understand pointer declarations and memory allocation
- Explain pointer operator manipulation
- Understand the differences between early and late binding
- Understand constant and non-constant pointer manipulation
- Software engineer dynamic memory allocation applications in C++

Data Structures

- Explain the similarities and differences between linked lists, trees, and vectors
- Software engineer examples of each data structure in C++
- Understand self-referential classes
- Understand queue and stack data structures
- Determine the proper implementation of each data structure type

Exception Handling

- Understand concept of exception-handling
- Explain other error handling techniques
- Understand rethrowing an exception
- Describe exception-handling constructors and destructors
- Describe processing unexpected exceptions
- Software engineer examples utilizing exception-handling