

COURSE OUTLINE

Beginning C++ with Game Programming

Course Description

IN 200. Beginning C++ with Game Programming. 3 hours credit. This course will enable the student to use C++ and the Standard Template Library to program text based games. The student will learn beginning procedural and object oriented programming using structured programming techniques to develop game applications. Before enrolling, the student should be competent in basic computer operation, use of the Internet for research and the use of standard office software like MS-Word, MS-Excel and MS-Access.

Course Relevance

In this course, the student will learn structured programming, algorithmic thinking and structured problem solving skills. These are skills necessary to develop useful, efficient, and entertaining games. This course also provides a foundation in object-oriented programming, which is necessary for success in more advanced programming courses and in the modern game development workplace.

Required Materials

Dawson, M., (2007). *Beginning C++ through game programming* (2nd ed.). Boston, MA; Thomson Course Technology.

Supplemental Materials

USB Flash Drive: 1 GB minimum

Learning Outcomes

The intention is for the student to be able to:

1. Convert gaming problems into programmable solutions using structured programming principles
2. Use the C++ Programming language to implement and test the solutions
3. Identify and use the basic elements of Structured Programming

Learning PACT Skills that will be DEVELOPED and/or documented in this course

Through the student's involvement in this course, he/she will develop his/her ability in the following PACT skill areas:

Technology Skills

1. Discipline-specific technology
 - By developing examples, exercises, and applications related to specific fields, the student will improve his or her understanding of the student's own field or other technological or vocational field

Major Summative Assessment Task(s):

These learning outcomes and the primary Learning PACT skills will be demonstrated by:

1. Preparation of a final project which assesses technical abilities.

Course Content

- I. Themes - Key recurring concepts that run throughout this course:
 - A. Translation of game design
 - B. Problem discovery, analysis and documentation process
- II. Issues - Key issues that will be addressed in this course: areas of conflict that must be understood in order to achieve the intended outcome:
 - A. Understanding the conflict between ideal game design and the development of a game practical application
 - B. Understanding the conflict between the development of a game practical application and real world constraints; money, time and manpower
- III. Concepts - Key concepts that must be understood to address the issues:
 - A. The translation of solutions into computer programs
 - B. The role of teams in project development
 - C. The importance of algorithmic thinking in problem analysis and structured programming development
- IV. Skills/Competencies - Actions that are essential to achieve the course outcomes:
 - A. Demonstrate fundamental computer concepts of C++ programming
 - B. Demonstrate the use of C++ Integrated Development Environment to design an application
 - C. Write programs in C++
 1. List the various data types
 2. Understand the precedence of arithmetic operators
 3. Write simple decision-making statements
 - D. Use basic problem-solving techniques
 1. Develop algorithms through the process of top-down, stepwise refinement
 2. Use selection structures and repetition structures
 - E. Create sub procedures and function procedures
 - F. Create arrays to store, sort, and search lists and tables of values
 - G. Manipulate characters in strings using various C++ functions

Learning Units

- I. Introduction to structured programming
 - A. History of programming languages
 - B. Compilers
 - C. Introduction to variables
- II. Basic elements of C++ IV using input/output
 - A. Types
 - B. Variables
 - C. Standard IO
 - D. Arithmetic operators

- III. Control structures
 - A. Booleans/Truth tables
 - B. Selection structures
 - C. Loops
 - D. Random numbers
- IV. Arrays and strings
 - A. String literals
 - B. Members of an array
 - C. Length of an array
 - D. Index of an array
- V. Standard Template Library
 - A. Functions STL
 - B. Includes directives and library extensions
- VI. Vectors
 - A. Relative and absolute values of vectors
 - B. Iterators
 - C. Algorithms
 - D. Methods
 - E. Functions
- VII. Passing values
 - A. References
 - B. Pointers
- VIII. Object oriented programming
 - A. Classes
 - B. Dynamic memory
 - C. Inheritance
 - D. Polymorphism
- IX. Design
 - A. TOE charts
 - B. Flowcharting
 - C. Psuedocode

Learning Activities

Independent and collaborative learning activities will be assigned to assist the student in achieving the intended learning outcomes. Learning activities will involve the student in the creation and design of game applications independently or in collaboration with others, and will include various aspects of design, technology, and project management.

Grade Determination

The student will be graded on assessment tasks, participation and the timely completion of class exercises and tutorials.