

GD345

C++ Programming for Game Developers

[Onsite]

Course Description:

This course provides an introduction to object oriented computer programming framed in the technical aspects of game programming. Students will apply the following concepts of variables, control structures, functions, arrays, data types, classes, inheritance and polymorphisms as they build a series of games.

Prerequisite(s) and/or Corequisite(s):

None.

Credit hours: 4

Contact hours: 50 (30 Theory Hours, 20 Lab Hours)

Syllabus: C++ Programming for Game Developers

Instructor: _____

Office hours: _____

Class hours: _____

Major Instructional Areas

1. Computer hardware, software, and memory
2. C++ syntax and semantics
3. Elements of game design and development
4. Graphics programming for games
5. Program development cycle
6. Components of C++ programming (constants, variables, functions, arguments)
7. Algorithms in programming
8. Control structures
9. Object-oriented programming
10. Game program compiling, testing and implementation

Course Objectives

1. Use proper C++ syntax in creating C++ programs for games.
2. Use numerical data in games programming.
3. Define functions to perform game programming tasks.
4. Produce images for games.

5. Describe the main types of control structures in C++ programming.
6. Develop animation for games.
7. Design a value-returning function for a game.
8. Explain how arrays and linked lists are used in game programming.
9. Incorporate strings within a game program.
10. Define object-oriented programming.
11. Create a game using C++ programming.

SCANS Objectives

SCANS is an acronym for Secretary's Commission on Achieving Necessary Skills. The committee, created by the National Secretary of Labor in the early 1990s, created a list of skills and competencies that the committee feels are necessary for employees to function in a high-tech job market.

1. Identify relevant facts and analyze information in a logical manner after locating and verifying information using resources and computers.
2. Identify common goals and examine all possible options for problem solving.
3. Identify problems, create and implement solutions, and revise solutions, as required.
4. Allocate time and energy for completing projects in a timely manner.
5. Exert a high level of effort and perseverance toward attaining goals.
6. Recognize problems and devise and implement a plan of action.
7. Demonstrate the ability to use authentic resources available, including the Internet, knowledge libraries, or other sources.
8. Locate, understand, and interpret information obtained from a variety of sources.
9. Identify the need for data; select, retrieve, and analyze information; and communicate the results of information analysis in written, graphical, and pictorial formats.
10. Compare and contrast two or more theories or alternatives to arrive at the best solution.
11. Apply procedures, tools, and equipment—including computers and related technologies—whenever required.

12. Evaluate alternatives and choose the best for a particular situation.
13. Use a systematic problem-solving process to analyze and solve a problem.

Course Outline

Note: All graded activities, except the Course Project, are listed below in the pattern of <Unit Number>.<Assignment Number>. For example, Lab 2.1 refers to the 1st lab activity in Unit 2.

| Unit | Activities |
|--|--|
| 1– Introduction to Computers and Games Graphics | <ul style="list-style-type: none"> • Content Covered: <ul style="list-style-type: none"> ○ <i>Starting Out with Games and Graphics in C++:</i> <ul style="list-style-type: none"> ○ Chapter 1, “Introduction to Computers and Programming,” pp. 1-29 ○ Chapter 2, “Graphics Programming with C++ and the Dark GDK Library,” pp. 37-75 • Assignments: 1.1 • Labs: 1.1 • Quizzes: 1.1 |
| 2– Numerical Data in C++ Programming | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++:</i> <ul style="list-style-type: none"> ○ Chapter 3, “Variables, Calculations, and Colors,” pp. 81-132 • Quizzes: 2.1 • Assignments: 2.1 • Labs: 2.1 |

| Unit | Activities |
|--------------------------------|---|
| 3– Functions | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 4, “void Functions,” pp. 139-173 • Quizzes: 3.1 • Assignments: 3.1 • Labs: 3.1 |
| 4– Graphics | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 5, “Working with Images,” pp. 179-208 • Quizzes: 4.1 • Assignments: 4.1 • Labs: 4.1 |
| 5– Control Structures | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 6, “Control Structures,” pp. 213-273 • Quizzes: 5.1 • Assignments: 5.1 • Labs: 5.1 |
| 6– Using Animation in Games | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 7, “The Game Loop and Animation,” pp. 281-339 • Quizzes: 6.1 • Assignments: 6.1 • Labs: 6.1 • Course Project: Assigned |

| Unit | Activities |
|---|---|
| 7– Adding Audio and Value-Returning Functions in Games | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 8, “The Vulture Trouble Game: Introducing Audio, Physics, and Text Effects,” pp. 345-403 ○ Chapter 9, “Value-Returning Functions and Mouse Input,” pp. 409-466 • Quizzes: 7.1 • Assignments: 7.1 • Labs: 7.1 |
| 8– Defining Arrays in Game Programs | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 10, “Arrays and Tile Mapping,” pp. 473-552 • Quizzes: 8.1 • Assignments: 8.1 • Labs: 8.1 |
| 9– Working with Strings and Files in Game Programming | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 11, “Strings and Files,” pp. 559-604 • Quizzes: 9.1 • Assignments: 9.1 • Labs: 9.1 |
| 10– Object-Oriented Programming for Games | <ul style="list-style-type: none"> • Read from <i>Starting Out with Games and Graphics in C++</i>: <ul style="list-style-type: none"> ○ Chapter 12, “Object-Oriented Programming,” pp. 611-676 • Quizzes: 10.1 • Assignments: 10.1 • Labs: 10.1 |

| Unit | Activities |
|----------------------|--|
| 11– Course Review | <ul style="list-style-type: none"> • Course Project: Submit |

Instructional Methods

The C++ Programming for Game Developers course introduces you to the theoretical and practical aspects of game programming and development. In this course, you will have the opportunity to use compilers and debugging programs as part of lab activities.

This course encourages active participation in classroom activities. Learning is enhanced by providing you the opportunity to participate in classroom discussions. You can appreciate your peers' suggestions, strategies, and findings on game programming. You will complete quizzes and labs based on classroom discussions.

There is one project in this course—a game that you will design and program using the concepts learned in this class.

The overall assessment strategy for this course includes quizzes, labs, homework assignments consisting of end-of-chapter questions, and the course project.

Instructional Materials and References

Student Textbook Package

Gaddis, Tony. *Starting Out with Games and Graphics in C++*. Reading, MA: Addison-Wesley, 2010.

Other Required Resources

In addition to the student textbook package, the following is also required in this course:

- Required software as outlined in Appendix A of *Starting Out with Games and Graphics in C++*

References

ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at <http://library.itt-tech.edu/> to access online books, journals, and other reference resources selected to support ITT Tech curricula.

Books

You may click “Books” or use the “Search” function on the home page to find the following book.

- Fullerton, Tracy, Christopher Swain, and Steven Hoffman. *Game Design Workshop: Designing, Prototyping and Playtesting Games*. Gilroy, CA: CMP Books, 2004.

Course Evaluation and Grading

Evaluation Criteria

The final grades will be based on the following categories:

| CATEGORY | WEIGHT |
|----------------|-------------|
| Quizzes | 20% |
| Assignments | 20% |
| Labs | 30% |
| Course Project | 30% |
| Total | 100% |

Grade Conversion Table

The final grades will be calculated from the percentages earned in the course, as follows:

| | | |
|---|---------|-----|
| A | 90-100% | 4.0 |
|---|---------|-----|

| | | |
|----|--------|-----|
| B+ | 85-89% | 3.5 |
| B | 80-84% | 3.0 |
| C+ | 75-79% | 2.5 |
| C | 70-74% | 2.0 |
| D+ | 65-69% | 1.5 |
| D | 60-64% | 1.0 |
| F | <60% | 0.0 |