Beginning C++ Programming for Games
Fall 2012 Syllabus

Course Number	CS 161C+	**Instructor**	Brian Bird
Credits	4	**E-mail**	BirdB@lanecc.edu
Classroom	Building 19 / Room 132	**Office**	19 / 152
Section	CRN 28044 Tu, Th 12:00–1:50 pm	**Office Phone**	541-463-3024
		Office Hours	M–Th, 4:00–5:00

Course Description

This course provides an introduction to programming using the C++ language in a game development environment. You will be exposed to the concepts and tools involved in C++ software development. Most of the programming assignments for the class involve relatively simple games. There are no explicit pre-requisites for the course, but you are expected to be familiar with general computer usage and the Windows environment.

Learning Outcome

The intent of this course is to enable you to write simple programs using proper C++ syntax and semantics, compile those programs, debug them as necessary, and run them.

Course Content

Technologies

C++	Code::Blocks IDE	

Themes and Issues

Software engineering	Algorithm development	C++ syntax
C++ semantics	Top down development	Object oriented development

Concepts

Constants, variables and scope	Data types	Program control structures
Interpreters and compilers	IDEs	Functions
Parameters	Libraries and linking	
Information hiding	Procedural vs. OOP	Classes and objects

Skills

Use Code::Blocks to enter, compile and run C++ programs.
Develop, test and debug algorithms using standard control structures.
Express algorithms in proper C++ syntax.
Implement, test and debug a variety of simple game programs in C++.
Describe concepts, themes and issues orally and in writing.

Learning Resources

Texts

Starting Out with C++: Early Objects, 7th Edition, by Gaddis, Walters and Muganda is the main text for the course. It is a good introduction to beginning programming concepts as well syntax of C++. It is also the text used for CS 162C+. Many of the lab exercises are taken directly from the text. An electronic copy of the source code for all of the sample programs in the text is available on the cd that came with the text. These files are also available on the web site for the textbook (http://www.aw.com/cssupport).

Simple Program Design: A Step-by-Step Approach by Leslie Anne Robertson will be used to introduce programming concepts and to practice the development of algorithms – the step by step set of instructions that describe how to accomplish programming tasks. There will be required reading in this book, but since the book will be available in the library you are not required to purchase it.

Web Sites

http://classes.lanecc.edu (LCC's moodle site) will be used to host the web site for the course. The site is designed to provide you with all of the information you need about the content and structure of the course including: lecture notes, sample files, assignment descriptions and due dates, practice and actual quizzes, and announcements related to the course.

Software

The hardware and software required for the course is available to all students in the CIT Main Lab on campus. You paid a fee when you registered for this course that provides you with unlimited access to CIT lab facilities.

Code::Blocks will be used as the C++ integrated development environment in class and in the CIT Main Lab. You can download Code::Blocks from http://www.codeblocks.org.

You may alternately use Visual Studio to complete your lab assignments. As a result of a partnership between Microsoft and many educational institutions, including LCC, students may obtain a copy of Visual Studio as well as other Microsoft Software free of charge. Your instructor will provide you with specific information about obtaining Microsoft software for educational use during the first week of the term.

Assessment Activities

The learning outcome will be demonstrated by completing lab assignments that progressively build on each other. Each lab will be assessed both by your peers and by the instructor, but your grade will only be determined by the instructor's assessment. These are the tentative lab assignments:

1. Lab 1 – Getting started in C++ and Code::Blocks.
2. Lab D – Practice with different representations of data
2. Lab 2 – Design, implement and test C++ programs using variables and input/output.
3. Lab 3 – Design, implement and test algorithms using selection
4. Lab 4 – Design, implement and test C++ programs using selection.
5. Lab 5 – Design, implement and test algorithms using repetition.
6. Lab 6 – Design, implement and test C++ programs using repetition.
7. Labs 7 and 8 – Design, implement and test C++ programs using functions.
8. Labs 9 – Design, implement and test a C++ classes.

In addition, there will be weekly reading quizzes and three exams.

Assessment and Grading

Specific grading criteria will be applied to each of the labs, quizzes, and exams you will be working on in this class. Part of the lab involves peer evaluation. I will provide you with lab evaluation worksheets for each lab.

The table below summarizes the possible points for each assessment task.

Assessment Activities	***Points for each***	***Number per Term***	***Total Points***
Labs & Homework	50	10	500
In-class exercises	10	10	100
Reading Quizzes	10	10	100
Exams	100	3	300
Course Total			***1000***

Letter grades for the course will be determined by the following percentages:

	-	no + or -	+
A	90 - 91	92 – 97	98 - 100
B	80 - 81	82 – 87	88 - 89
C	70 – 71	72 – 77	78 - 79
D	60 - 61	62 – 67	68 - 69
F	Below 60		

Late Work

- The grade for labs submitted after the due date will be reduced by 10%. Grades for labs submitted after the in-class evaluation will be reduced by 25%.

- Quizzes and exams cannot be taken after the due date.
- Exceptions will only be made for severe illness or emergency situations.

Course Schedule

Term Schedule (tentative)

Week	***In Class Lecture /Learning Activities***	***Things To Do Outside Class***
1 9/23	• Introduction to the course, to C++ • Introduction to Code::Blocks • Constants and Variables	• Read Gaddis et al chapters 1 and 2 • Chapter 2 quiz • Lab 1, first C++ Projects.
2 9/30	• Machine Representation of data • Data types	• Read the data representation tutorial • Take the data representation quiz • Lab D, data representation
3 10/7	• Input and Output • Statements and expressions	• Read Gaddis ch 3, Expressions & Interactivity • Chapter 3 quiz • Lab 2, interactivity
4 10/14	• Algorithm design process • General control structures: sequence & selection	• Read Robertson ch 1 - 4 • Read the lecture notes thoroughly • Machine Representation of Data Quiz • Lab 3, sequence & selection algorithms
5 10/21	• C++ control structures: sequence & selection • **Exam 1**	• Read Gaddis ch 4, Making Decisions • Chapter 4 quiz • Lab 4, Selection in C++ • Study for the first exam.
6 10/28	• Control structures in algorithms: repetition	• Read Robertson ch 5 • Read the lecture notes thoroughly. • Algorithm quiz • Lab 5, repetition algorithms
7 11/4	• Control structures in C++	• Read chapter 5, Looping. • Chapter 5 quiz • Lab 6, Repetition in C++. • Study for the second exam.
8 11/11	• Functions • User defined functions • **Exam 2**	• Read Gaddis ch. 6, Functions. • Chapter 6a quiz • Lab 7, Functions.
9 11/18 Thanksgiving holiday: 11/20	• Parameters, return values and scope	• Chapter 6b quiz • Lab 8, Multi-function programs.
10 11/25	• Classes and Objects	• Gaddis ch. 7, Classes and Objects • Chapter 7a quiz • Lab 9, Car class.
11 12/2	Finals Week • **Exam 3**	

Academic Honesty

While students are encouraged to discuss labs and quizzes and to use each other as resources, each student is responsible for his/her own work. In other words you can help each other, but you can't copy someone else's work. In addition, each individual student must complete their own lab assignment, not a group of students working together.

Academic Calendar for Fall Term 2012

Term begins	9/24/12
Last day to receive a refund	9/30/12, 11:59 p.m.
Veterans Day holiday	11/12/12
Last day for schedule changes	11/16/12
Thanksgiving vacation	11/22/12–11/25/12
Finals week	12/3/12–12/8/12
Term ends	12/8/12
Winter break	12/9/12–1/6/13

Disability Services

If you need support or assistance because of a disability, you may be eligible for academic accommodations through Disability Services. For more information, contact Disability Services at 463-5150 (voice) or 463-3079 (TTY), or stop by building 1, room 218.