

AUSTIN COMMUNITY COLLEGE

DEPARTMENT OF COMPUTER STUDIES AND ADVANCED TECHNOLOGY

Spring 2016 Course Syllabus

COSC 1337 – Programing Fundamentals II (C++)

Instructor:

Office Telephone:

Office:

Office Hours:

E-mail:

Course Description/Rationale

Course Description: This course is an introduction to the C++ programming language and its subset, the C programming language. Program structure, block, storage types, console and file I/O, functions, arrays, strings, pointers, call-by-reference, call-by-value, and dynamic memory allocation will be discussed. The concept and use of classes will be covered in some detail. The differences between C++ and C will also be discussed.

Pre-requisite: COSC 1336 or departmental approval

TSI Skills: Reading and Math

Instructional Methodology: This course will be taught over the Internet. The course material will be covered in reading assignments from the textbook supplemented by some online materials. Students will need to complete programming assignments on their own computer, or they will need to visit one of the Computer Studies open labs. Exams will be administered in the ACC Testing Centers.

Course Rationale: This course is designed to teach students the C++ programming language and introductory and intermediate programming concepts with examples and applications using the C++ language. The course builds and extends topics covered in the prerequisite course,

COSC 1336 and prepares students for more advanced programming courses such as ITSE 2331 (Advanced C++ Programming) and COSC 2436 (Programming Fundamentals III: Data Structures) as well as for entry level programming employment. The course is required for an Associate Degree in several Computer Information Systems and Computer Science degree areas.

Student Learning Outcomes/Learning Objectives

Course Objectives/Learning Outcomes:

1. Demonstrate a thorough understanding of modular programming by designing programs that require the use of programmer-defined functions.
2. Demonstrate a thorough understanding of arrays by designing and implementing programs that search and sort arrays.
3. Demonstrate a thorough understanding of the object-oriented programming concepts of encapsulation, data abstraction and composition by designing and implementing classes including the use of overloaded functions and constructors.
4. Demonstrate a thorough understanding of the concept of pointers and dynamic memory allocation by designing and implementing programs using pointers and dynamic memory allocation.
5. Demonstrate a thorough understanding of the implementation of programmer-defined functions and classes by writing code, performing unit testing and debugging of multiple complex programs.
6. Demonstrate good documentation style in all of the programs written in this course.
7. Demonstrate proficiency in implementing data validation code, performing unit testing, and developing test plans while implementing robust solutions to the assignments in this course.
8. Demonstrate a thorough understanding of stream input/output for both console and files.
9. Demonstrate an understanding of the differences between C and C++ in the areas of strings, pass by reference/passing pointers, and structs by designing and implementing programs that use C strings, C++ strings, C language structs and classes.

Competencies

1. Develop programs using procedural and object-oriented programming concepts.
 - 1.1.Design, code and test programs that use input, output and arithmetic operations.
 - 1.2.Design, code and test programs that use branching and looping control structures.
 - 1.3.Design, code and test modular programs that contain multiple functions.
 - 1.4.Design, code and test programs that contain programmer-defined data types (classes).
2. Develop programs using C structs, classes and arrays.

- 2.1.Design, code and test programs that use arrays and structures (arrays of structures).
- 2.2.Design, code and test programs that use two-dimensional arrays.
- 2.3.Design, code and test programs that use arrays and sequential search.
- 3. Develop programs using pointers, dynamic memory allocation, files and strings.
 - 3.1.Design, code and testprograms that use pointers, dynamically-allocated arrays and sorting.
 - 3.2.Design, code and test programs that use C++ strings and input and output files.
 - 3.3.Design, code and test programs that use C strings

Scans Competencies:

SCANS competencies have been identified that are relevant to the level of instruction in the community college environment. These competencies reflect the knowledge and skills employees need to succeed in any occupation. To be successful in this course the following SCANS competencies will need to be applied:

RESOURCES 1.1 Manages Time	INTERPERSONAL	INFORMATION 3.3 Uses Computers to Process Information	SYSTEMS
TECHNOLOGY 5.2 Applies Technology to Task	BASIC SKILLS 6.3 Arithmetic 6.5 Listening	THINKING SKILLS 7.5 Knowing How to Learn	PERSONAL SKILLS

Readings

Approved Course Text:

You can use any recent edition of Starting Out with C++ Early Objects by Gaddis et. al. I recommend the 7th or 8th edition:

Starting Out with C++ Early Objects 7th Edition, Gaddis, Walters, Muganda, Addison Wesley,
ISBN-13: 9780136077749

OR

Starting Out with C++ Early Objects 8th Edition, Gaddis, Walters, Muganda, Addison Wesley,
ISBN-13: 9780133360929

In addition, students will need, separately, to purchase a license for CodeLab (a web-based interactive programming exercise system) at a cost of around \$25.00. I will have instructions on how to purchase this posted in Blackboard.

Course Requirements

Grade Policy:

Grade will be assigned based both on concepts and practical application. Exams, exercises, and programming assignments will be a part of the grade. An overall grade will be assigned on the following grading scale.

The course grade will be based on 3 semester exams, 10 programming assignments and exercises.

Component	Weight
Exercises (CodeLab)	6.5 %
10 programming assignments program 1 @ 1 % programs 2 - 10 @ 3.5 % each	32.5 %
3 semester exams Exams 1 & 2 @ 20 % each Exam 3 @ 21 %	61 %
Total	100 %

Grade Scale:

90 % - 100 %	A
80 % - 89 %	B
70 % - 79 %	C
60 % - 69 %	D
0 % - 59 %	F

A student must score 60% or above to pass the course.

Course/Class Policies

Academic Integrity

A student is expected to complete his or her own projects and tests. Students are responsible for observing the policy on academic integrity as described in the current Student Policies Handbook.

The penalty assessed will be in accordance with the current policy.

Incomplete

A student may receive a temporary grade of “I” (Incomplete) at the end of the semester only if ALL of the following conditions are satisfied:

1. The student is unable to complete the course during the semester due to circumstances beyond their control.
2. The student must have earned at least half of the grade points needed for a “C” by the end of the semester.
3. The request for the grade must be made in person at the instructor’s office and necessary documents completed.
4. To remove an “I”, the student must complete the course by two weeks before the end of the following semester. Failure to do so will result in the grade automatically reverting to an “F”.

Freedom of Expression Policy

It is expected that faculty and students will respect the views of others when expressed in classroom discussions.

Tutoring

Free tutoring is provided for this course. For schedules and details please refer to [Tutoring Schedule](#).

ACC Distance Learning Support Services

The Help! For Distance Learning Students page contains support links and reference information that students need to successfully complete a Distance Learning course:

[Help! For Distance Learning Students](#)

Testing Center Policy

<http://www.austincc.edu/testctr/>

For student information see:

<http://www.austincc.edu/support-and-services/services-for-students/testing-services/instructional-testing/testing-center-guidelines>

Attendance / Withdrawal

Students who do not complete orientation during the first week of class and do not contact the instructor will be withdrawn from the class.

Students are expected to regularly monitor their ACC email and to keep up with the course schedule. Students who are more than 2 weeks late in submitting assignments or exams, and who do not respond to email from the instructor may be withdrawn from the class.

Students who do not comply with course policies may be withdrawn from the class.

It is the student's responsibility to complete a Withdrawal Form in the Admissions Office if they wish to withdraw from this class. **According to the college calendar, the last day to withdraw for 16 and 12 week classes is Monday, April 25.** It is not the responsibility of the instructor to withdraw the students from their class even though the instructor has the prerogative to do so under the above listed circumstances.

Students who enroll for the third or subsequent time in a course taken since Fall 2002 are charged a higher tuition rate. State law permits students to withdraw from no more than six courses during their entire undergraduate career at Texas public colleges or universities. With certain exceptions, all course withdrawals automatically count towards this limit. Details regard this policy can be found in the ACC College Catalog.

Student Files – Privacy

The information that a student stores in his/her student volume in the Computer Studies Labs may be viewed by their instructor for educational and academic reasons.

Students with Disabilities

Each ACC campus offers support services for students with documented physical or psychological disabilities. Students with disabilities must request reasonable accommodations through the Office for Students with Disabilities on the campus where they expect to take the majority of their classes. Students are encouraged to make this request three weeks before the start of the semester. (Refer to the current ACC Student Policies).

Communication

Most communications for this course will be by Piazza or Blackboard.

We will be using Piazza for class discussion. The system is designed to make it easy to get help from classmates, the grader and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at:

Students are welcome to call or come by my office during office hours. If you plan to come by my office, it is best to contact me the day before to be sure that I do not have a meeting that conflicts with my office hours. If you need to see me in person and cannot come during my office hours, please contact me so that we can arrange a time to meet. If you need to contact me outside office hours, it is best to email me.

I normally check Piazza and email multiple times each day, including weekends. However, sometimes I will miss a day on the weekend. I try to respond within 24 hours (usually faster).

Grades will be posted in the ACC Blackboard system:

<http://acconline.austincc.edu>

Students should use their ACCmail accounts for class communications.

<http://www.austincc.edu/accmail/>

Course materials including assignments, assignment due dates, exam reviews and exam due dates will be posted in Blackboard. Some of these materials may also reside on my course web site at:

<website>

All students are expected to check the class Blackboard site, Piazza and their ACCmail accounts on a regular basis (preferably every couple of days but at least once a week).

For information on how to log onto Blackboard and ACCmail please visit the following sites:

<http://irt.austincc.edu/blackboard/StudentSupport.php>

<http://www.austincc.edu/google/>

Safety Statement

Each student is expected to learn and comply with ACC environmental, health and safety procedures and agree to follow ACC safety policies. Emergency posters and Campus Safety Plans are posted in each classroom. Additional information about safety procedures and how to sign up to be notified in case of an emergency can be found at

<http://www.austincc.edu/emergency/>.

Anyone who thoughtlessly or intentionally jeopardizes the health or safety of another individual will be immediately dismissed from the day's activity, may be withdrawn from the class, and / or barred from attending future activities.

Software

You must have access to a computer with Internet access and a C++ compiler. You may also use the computers in the Computer Studies labs on any of the main ACC campuses. In addition, the Computer Studies Department has an agreement with Microsoft Corporation that allows students to obtain compilers at a greatly reduced cost. Contact your instructor for more information.

Orientation

Orientation is required. The Orientation instructions are on the Blackboard site for this class. Students who do not complete orientation during the first week of class and do not contact the instructor will be dropped from the class.

Class progress

Students are expected to keep up with the schedule of the class. In this course, each new concept builds on previous concepts. Regular progress through the material is crucial to success on tests and assignments.

Participation

Students are expected to participate in discussions. Questions and comments of general interest should be sent to the class Google group so that everyone can benefit from the discussion. Other questions, such as questions about your grades, should be sent directly to the instructor. Students are encouraged to ask questions. There are no "dumb" questions! The instructor knows that the material is difficult to grasp at times and will be glad to give additional examples or one-to-one assistance when asked.

Programming Assignments

Success in this course depends on the ability to successfully complete the assigned homework. Students are expected to do the assignments, which are designed to provide experience and practice with the concepts and techniques covered in the course.

Submitting Assignments: Programming assignments are located in Blackboard. Assignments must be submitted by clicking the assignment link in Blackboard and attaching the program source code to the assignment. **Do not submit project files, make files, object files, etc. I only want you to submit files that contain C++ code that you wrote.**

Deadlines: Programming assignment due dates will be posted on the Course Schedule in Blackboard. If you miss the due date, you are running behind and may have trouble completing the course. **The last day that I will accept programs 1 through 8 is Sunday, May 8 at 11:59 pm. Programs 9 and 10 must be turned in by Friday, May 13 at 11:59 pm.**

Assignment Grading: I try to grade programs within one week of submission. Assignments will be graded on the following basis:

- 20 % - Documentation - proper use of variable names, indentation, comments, etc.
- 80 % - Program operates correctly with instructor input data and performs all required functions

Program documentation involves the proper use of variable names, indentation, comments, etc. These conventions are described in the textbook. I will post a summary of these guidelines on my web page. It is very important to make the programs as readable as possible and the instructor will strictly enforce the documentation guidelines for this course.

Please contact me if you have questions about the assignment expectations or grading process.

Exams

The material covered in each chapter of the textbook builds on material covered in previous chapters. For this reason, the second and third exams will include material from previous exams. If a mistake is made related to material covered on a previous exam, points will still be deducted. Exams will cover material from the textbook and supplemental material from the class web site. Topics for each exam will be listed on the **Chapter Review** web pages.

All exams will be administered at an ACC Testing Center. Testing facilities are located at all major campuses and are open throughout the week. Check the current schedule for the testing facility you wish to use. Allow at least two hours for each exam.

You may bring up to 3 pages (8.5 x 11" maximum) of notes for each exam. You may write on both sides of the page. The notes will be attached to the test and sent to the instructor, so you will want to make a copy before you take the exam. I try to grade exams within two weeks of receipt. After your exam is graded, I will email you with comments on items that were missed. If you want to review and discuss the test with the instructor you may do so by scheduling an appointment with the instructor. Exams are not returned to the student.

Deadlines: I will post due dates for exams on the Course Schedule in Blackboard. If you miss the exam due date, you may have trouble completing the course. **The last day that I will accept exams 1 & 2 is Sunday, May 8. Exam 3 will be available at all Testing Centers through Wednesday, May 11. Exam 3 will be available at the NRG CIS Lab (4232) and the RVS CIS Lab (RVS G9141) Thursday and Friday, December 12 & 13.**

Course Subjects

Programming Fundamentals II C++

Typical 16 and 12 week Course Schedules are included below.

Typical 16 week Course Schedule

Week	Topic
1	Chapter 1 - Introduction to Computers and Programming Chapter 2 - Introduction to C++ Chapter 3 - Expressions and Interactivity
2	Chapter 4 - Making Decisions (Selection) Chapter 5 - Looping (Repetition)
3	Chapter 6 - Functions
4	Chapter 7 - Introduction to Classes and Objects
5	Chapter 7 (continued)
6	Exam 1
7	Chapter 8 - Arrays
8	Chapter 8 (continued)
9	Chapter 9 – Searching, Sorting and Algorithm Analysis
10	Chapter 10 – Pointers
11	Exam 2
12	Chapter 11 - More about Classes and OOP (selected topics)
13	Chapter 12 - More about Characters, Strings, and the string Class
14	Chapter 13 - Advanced File and I/O Operations
15	Chapter 14 - Recursion (selected topics) Chapter 17 - Linked Lists (selected topics)
16	Exam 3

Typical 12 week Course Schedule

Week	Topic
1	Chapter 1 - Introduction to Computers and Programming Chapter 2 - Introduction to C++ Chapter 3 - Expressions and Interactivity
2	Chapter 4 - Making Decisions (Selection) Chapter 5 - Looping (Repetition)
3	Chapter 6 - Functions Chapter 7 - Introduction to Classes and Objects
4	Exam 1
5	Chapter 8 - Arrays

6	Chapter 9 – Searching, Sorting and Algorithm Analysis
7	Exam 2
8	Chapter 10 – Pointers
9	Chapter 11 - More about Classes and OOP (selected topics)
10	Chapter 12 - More about Characters, Strings, and the string Class
11	Chapter 13 - Advanced File and I/O Operations Chapter 14 - Recursion (selected topics)
12	Chapter 17 - Linked Lists (selected topics) Exam 3

Note: The instructor has the prerogative to change the course schedule as required.