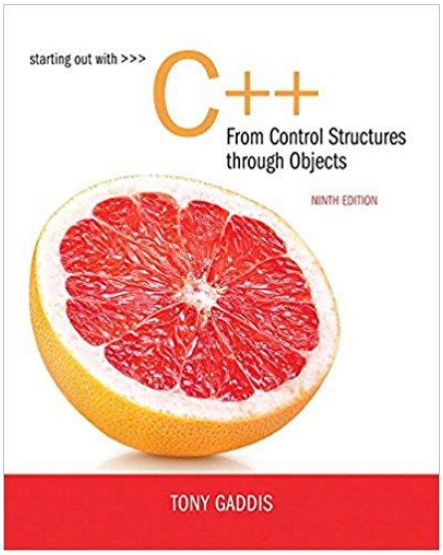
Syllabus

CSC/CIS 17A C++ Programming

General Course Information	
1	**Contact Information** Instructor: Dr. Mark E. Lehr Email: mark.lehr@rcc.edu Phone: (951) 222-8260
2	Course Description **Course Sequence of Study** (Course content related to the Gaddis textbook sections/chapters) 1. Review of C++ concepts. Chapters 1-8 2. Introduction and overview of object-oriented programming and C++. Chapter 13 and 14. 3. Functions Chapter 6 and Chapter 8 4. Pointers Chapters 7, 9 and 10 5. Structures Chapter 11 6. Classes and members Chapter 13

	7. Polymorphism Chapter 15 8. Constructors and destructors Chapter 14 9. Friends Chapter 14 10. Overloading Chapter 6 and 13 11. Inheritance Chapter 15 12. Advanced input and output Chapter 12 13. Templates Chapter 16 14. Linked Lists Chapter 17 15. Stacks and Queues Chapter 18 16. Recursion Chapter 19 17. Binary Trees Chapter 20
3	**Textbook:** (Yes, if you have an earlier version such as the 8th edition, you will be able to use it in my class, however, if you have an opportunity to obtain the new 9th edition then please do.)  **Starting Out with C++ : From Control Structures through Objects**

	• **Loose Leaf:** 1344 pages • **Publisher:** Pearson; 9 edition (February 27, 2017) • **Language:** English • **ISBN-10:** 0134443829 • **ISBN-13:** 978-0134443829
4	**Student Learning Objectives**: 1. Analyze and understand the Object-Oriented C++ environment. 2. Demonstrate the use of specialized terminology, directives, and features of the C++ language. 3. Apply theoretical business, scientific, and mathematical concepts in writing and executing programs in the C++ language using Object-Oriented programming methodology. 4.Demonstrate specified problem-solving skills using the C++ language. Bottom Line - At the end of the course you will need to demonstrate the ability to create 2 dimensional dynamic arrays of objects that utilize abstract, inherited, and generic programming concepts.

5	**Course Requirements** (assignments as a percentage of grade) Homework -- 20% Assigned at the end of every chapter. Graded as pass/fail for each problem. Projects -- 40% Two projects are required including a written report, full documentation, as well as working code. The first project is worth 20% and the final project is worth 20%. Midterm -- 20% Appropriate problems will be assigned which tests the students knowledge of the subject matter. Requires commented working code. Final -- 20% Will test the students ability to solve problems utilizing all the constructs covered in class. Also, requires commented working code. Note: All assignments, projects, and exams are to be done individually by the student. The homework maybe discussed but each student is responsible for turning in their own assignments.
6	Grading Rubric
7	**Grade scale for entire course** 90 - 100 A 80 - 89 B 70 - 79 C 60 - 69 D 0 - 59 F

8	**Course Homework Assignments** Lessons — Homework Problems 1 – Chapters 1-4 — 3.12, 3.13, 4.10 and 2 other problems 2 – Chapters 4-8 — 5.11, 6.7, 7.6, 8.7 and 1 other problem 3 – Chapter 9 — 9.2, 9.6, 9.7 and 7 other problems 4 – Chapter 10 — 10.4, 10.6 and 3 other problems 5 – Chapter 11 — 11.9 and 9 other problems 6 – Chapter 13.1-9 — 13.1,13.4 and 3 other problems 7 – Chapter 13.10-18 — 13.5,13.6 and 3 other problems 8 – Chapter 12 — 12.7,12.8, 12.11 and 2 other problems 9 – Chapter 14.1-4 — 14.1,14.2 and 3 other problems 10 – Chapter 14.5-8 — 14.3,14.4 and 3 other problems 11 – Chapter 15 — 15.1, 15.4, 15.6 and 7 other problems 12 – Chapter 16 — 16.1, 16.2, 16.3, 16.4, 16.5 and 5 Additional homework that might utilize lists, recursions and trees
9	**Course Structure** – Expectations No late homework assignments, projects, or exams! No makeup projects, tests, or finals! Consideration will be given to students that are going out of town on business for the week, etc... Must make prior arrangements with the instructor. Assignment due dates are posted in Blackboard. **Lab Hours** Online -> Lab assignments will be provided. A total of **18** laboratory hours are **required to pass** this course. The first hour is to be recorded by the end of the second week of class.

	The 9th is to be completed prior to the 9th week of class. All 18 hours must be completed prior to the final. An F will be recorded for the class grade if less than 18 hours of lab are completed prior to the end of the semester.
10	**Statement on Accommodations** If you have a physical, psychiatric/emotional, medical, or learning disability that may impact your ability to carry out assigned course work, I would urge that you contact the staff in DSP&S. They will review your concerns and determine, with you, what accommodations are necessary and appropriate. All information and documentation is confidential.
11	**NET-iquette** Like being in a face-to-face class, students in an on-line class must be able to discuss and debate divergent views without ridicule or personal attack. An important part of learning is considering the broad range of views possible on any one subject. You will be engaging in on-line discussions on topics that may yield a diverse array of opinion. Any behavior that is considered offensive in a classroom setting will be considered offensive on-line. This includes, but is not limited to the use of profanity, racial, sexual, or religious epithets, harassing or disrespecting another person on-line. Remember, all discussions are monitored daily.

Course Requirements

Follow the 16-week sequence of study in the sections of the on-line course syllabus. (Note: the 6 & 8 week intersession cover the same material) Complete assignments and post those results requested.

Note: See Assignments section for complete description of each course requirement. You will earn points for completion of course requirements for the PROJECTS as described below:

CSC/CIS 17A
Grading Scale for **Programming** Elements of

Projects, Presentations

Assignments - **On a scale of 1 to 10**

9 - 10 points

- specifically addresses all parts of the Assignment
- thoughtful, clearly commented and well organized code
- demonstrates keen understanding of overall lesson content
- integrates content of recommended programming components
- shows relevancy to lesson content
- raises additional questions or issues not specifically covered in class
- includes justification and/or documentation to support results
- includes programming content from other Chapters not covered in class with full documentation flowcharts/doxygen.

7 - 8 points

- addresses Assignment in less specific terms
- may not always address all parts of the program
- code not always supported or justified
- demonstrates good understanding of overall lesson content
- may include digressions from lesson content
- some integration of information
- Minimal to inadequate flowcharting/doxygen.

5 - 6 points

- addresses Assignment general terms
- does not address all parts of the Assignment
- code frequently not supported or justified
- demonstrates some understanding of overall lesson content
- includes digressions from lesson content
- organization and focus are inconsistent

3 - 5 points • organization and focus are poor • does not address most parts of the Assignment • code not supported or justified • demonstrates minimal understanding of overall lesson content
Less than 2 points • lacks organization and focus • shows no understanding of lesson content • fails to address any part of the Assignment • no unique application of concepts

Grading

Your grade in this class will be based on your performance on assignments, projects, exams, and your participation in course learning activities. You will earn points for each course requirement completed; the number of points you earn determines the letter grade you receive according to the following scale:

Percentage	Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	F

Class Information Resources

If you find internet resources related to course content on your own, please share that information with the class by posting it on the Class Discussion Board.

COURSE STRUCTURE

How to Study

You should plan to devote at least 12 hours per week during a normal semester to complete the coursework/lesson for the week. For inter-sessions, this goes up dramatically to 24 to 36 hours a week. Establish a study schedule that works for you by setting aside a specific time to study when you are free of distractions and other commitments.

STEP 1: READ THE CLASS SYLLABUS.

The Syllabus is your "roadmap" to the class. It contains a description of the course, an overview of each element of this course, a list of the course materials, and requirements, and an introduction to your instructor. You should review the class syllabus frequently and refer to it first when you have a question about the class.

STEP 2: STUDY THE DAILEY LESSON.

Here you will find the learning objectives, the content covered for the week, a list of the background material you will need to explore and the assignments. Each lesson in the 6 or 16-week class follows the same format designed to present content and assignments in the most effective manner.

STUDY BACKGROUND MATERIAL provided in the lesson.

ASSIGNMENTS

Suggested sequence of Study

Given there are roughly16 lessons and 16 weeks to complete the course of study here is a suggested review sequence that will be useful each week:

Dailey Assignments

In the **EVALUATE** section of each lesson, you will normal write programs covering the lesson's content. The assignment will be submitted back to me for evaluation. Once the time has expired, then turn in what you have and proceed to the next lesson. (Note: I always make the online assignments open 24 to 48 hours past this time in case there are server problems. I wouldn't take regular advantage of this.)

Mid-Term and Final Exams

These will be problems designed to test your knowledge of the material. Normally you will be given a day or two to complete the Exams. I want them run in Netbeans. You can choose to use other IDE's but Netbeans is what I will use. Fail to test them in Netbeans, and your grade will be impacted!

First and Final Project

These two projects are major assignments. There is a written part as well as a coded part. Just as in the homework assignments and exams, you will turn in files that I will run to evaluate your code. The first project is a minimum 250+ line program that you will write to solve a particular problem you are interested in. These are of your own choosing. The Final project is a minimum

1000+ line program that again is of your choosing but must solve a particular problem of interest. Examples will be referred to and suggestions will be given if you are having trouble determining what would be appropriate. Prior to working on the project, you will gain ideas from your classmates in the discussion board. You will also provide a proposal to turn in for my approval. A written report will be required when the projects are due as well as running code.