Riverside City College

CSC/CIS 5 - Programming Concepts and Methodology I: C++

Synopsis → We will cover chapters 1 to 8 as core concepts. 10 problems/Chapter at a pace of roughly 2 Chapters/week. Chapters 9, 11 and 13 will also be discussed and examples/problems assigned.

Homework / Midterm / Project1 / Final / Project2 each count for 20% of the grade. ½ content of course.

Labs make up the other course content without which you will not be able to do the above assignments.

Instructor: Dr. Mark E. Lehr

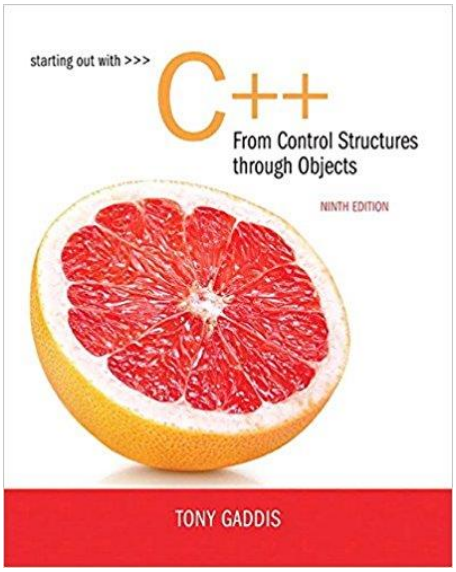
E-Mail: mark.lehr@rcc.edu

Office Hours: hyper_link

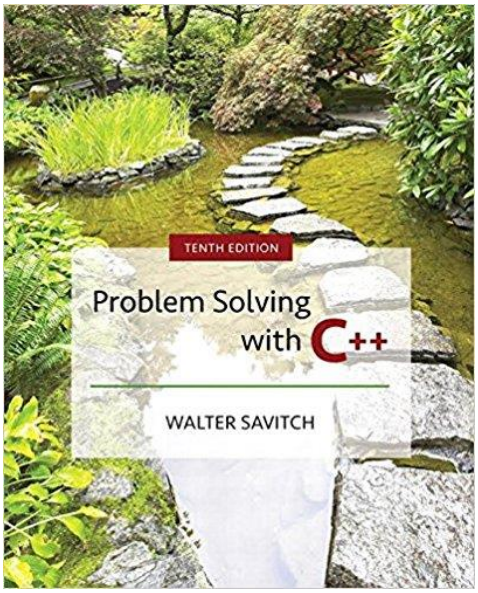
Online Hours: Tues-Thurs 8 to 8:30PM → https://zoom.us/join type meeting ID = 437-208-1701

Outline: hyper_link

Required Text, either not both unless you are really into programming and in that case you should also get the Deitel book on C++ as well.



- **Paperback:** 1344 pages
- **Publisher:** Pearson; 9 edition (February 23, 2017)
- **Language:** English
- **ISBN-10:** 0134498372
- **ISBN-13:** 978-0134498379



- **Paperback:** 1120 pages
- **Publisher:** Pearson; 10 edition (February 20, 2017)
- **Language:** English
- **ISBN-10:** 0134448286
- **ISBN-13:** 978-0134448282
- **Product Dimensions:** 7.3 x 1.6 x 8.8 inches

Required Materials

Netbeans for all projects/assignments/tests/and labs in the class.

Course Description

Introduction to the discipline of computer science incorporating problem definitions, algorithm development, and structured programming logic for business, scientific and mathematical applications. The C++ language will be used for programming problems. 54 hours lecture and 54 hours laboratory.

Prerequisite - Advisory

CIS 1A – Introduction to Computer Information Systems. It is advised that before entering this course students be able to:

1. Identify the fundamental computer concepts and terminology used for input, processing, output, and storage.
2. Identify the key features of a variety of software such as operating systems, word processors, spreadsheets, databases, communications and graphics.
3. Create electronic presentations with presentation graphics.
4. Demonstrate the principles of Internet research.
5. Understand the principles of computer security, ethics and privacy

Course Objectives

Upon successful completion of the course, students should be able to:

1. Describe the software development life-cycle
2. Describe the principles of structured programming and be able to design, implement and test structured programs.
3. Explain what an algorithm is and its importance in computer programming.
4. Summarize the evolution of programming languages illustrating how this history has led to the paradigms available today.
5. Use pseudo-code, flowcharts, and a programming language to implement, test, and debug algorithms for solving problems. Identify the information requirements, synthesize the algorithmic steps needed to transform the data input into the required output information, and organize the output format to facilitate user communication.

6. Demonstrate different forms of binding, visibility, scoping, and lifetime management.

7. Create computer programs using the principles of structured programming and demonstrate the use of an IDE with appropriate libraries. Design, implement, test, and debug programs that use fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and functions.

8. Apply the principles of logical and programming concepts to develop solutions for gaming, business, scientific and mathematical problems.

Lab Hours

A total of **54** laboratory hours are **required to pass** this course. Labs are scheduled immediately after the class lecture. Each hour of Lab missed subtracts 2 per cent from your grade. Only 2 lab sessions can be made up during the semester. There are **no exceptions**. Lab hours are in **addition** to the scheduled class hours.

The online class will have lab assignments posted in Blackboard consisting of the 54 hours we would normally do in class.

School and Class Rules

Attendance and participation: If you do not attend you cannot participate; if you do not participate, you cannot learn and therefore will not pass this class. You must show up to class prepared and ready to learn. Daily attendance will be taken and has same requirements as lab. For those in the online class, there will be discussion posts and assignments that will count as participation.

Classroom decorum: Be on time to class. If the door is locked and class has started, you missed class for that day. No food or drink is allowed in the classroom. You are expected to be cooperative and respectful during class. Using your cell phone, disruptive talking or behavior is considered rude and you will be asked to leave if you insist on being rude; you will not receive points for that day. If you have an emergency and must use your cell phone, please step out of the classroom and return when you are done. You **do not** have the right to disrupt the learning of others.

Online decorum: Be on time posting to the discussion board. Assignments are opened and have a finite amount of time to post responses or solutions. You are expected to be cooperative and respectful during the discussion board or online office hours. You **do not** have the right to disrupt the learning of others.

Statement on Academic Dishonesty:

RCC defines plagiarism as, “Presenting another person’s language (spoken or written), ideas, artistic works or thoughts as if they were one’s own.” This includes using someone elses C++ code. Plagiarism is academically dishonest. Students must make appropriate acknowledgment of

the original source where material written or compiled by another is used." Cheating or dishonest practices, such as turning in the writing of someone else and claiming it as your own, will result in your receiving a failing grade on the assignment and possibly for the course.

ADA Information

Please let me know if you need accommodations for a documented disability. The office of Services to Students with Disabilities will also be able to provide help and assistance.

Course Activities and Class Format

Daily classroom instruction will consist of lectures, discussions, and demonstrations, as well as hands on work, both collaboratively and individually. Lecture will be delivered verbally, supported by PowerPoint presentations, chalk board drawings, and on occasion, paper handouts, among other methods. Periodically, students will be required to interact and work in groups or teams to reinforce learning. All lab, homework, and class projects will use Git/Github. Labs will use the class site Github repository where students are collaborators. Homework and projects will use student repositories with the instructor as the collaborator.

Lab: Flowcharts/pseudo-code/programs per assigned topic. Students may collaborate on solutions and coding. Peer talks will also be required discussing the lecture content.

Homework: 10 flowcharts/pseudo-code/programs per assigned topic. Students may collaborate on homework solutions, but the code must be your own.

Mid-Term: No makeups. 5-6 or more computer programs which includes a menu. No collaboration allowed.

Project I: A game of your choice to illustrate all the concepts covered up to the Midterm. Full documentation with a write-up including flowcharts/pseudo-code. Your unique solution. Minimum of 100 lines of code. Unused, or duplicative code, and unnecessary i/o will not count in the lines of code. Minimum lines of code and write-up will produce a C grade or less. This is one place to impress me with the knowledge and expertise you acquire in class.

Final Exam: No makeups. 6 or more computer programs which includes a menu. No collaboration allowed.

Project II: A game of your choice to illustrate all the concepts covered up to the Final. You may choose your first project as the basis for this project. Full documentation with a write-up including flowcharts/pseudo-code. Your unique solution. Minimum of 250 lines of code. Unused, or duplication code, and unnecessary i/o will not count in the lines of code. Minimum lines of code and write-up will produce a C grade or less. This is one place to impress me with the knowledge and expertise you acquire in class. This project will be presented on Finals day.

Reading Assignments: Reading assignments will be given and could be followed with impromptu quizzes which will be graded pass/fail and count as your attendance for lecture or lab.

Blackboard

This course includes the use of Blackboard. It is the student's responsibility to visit Blackboard and stay current with respect to assignments and grades. Blackboard will be used in the following ways:

Course materials: Course materials such as the syllabus, homework assignments, and review materials will be posted on Blackboard. Reading assignments will be posted on Blackboard as well.

Homework/Projects/Exams: will be posted from Blackboard.

Announcements: In addition to making announcements in class, all announcements will be posted on Blackboard.

Emails: All emails sent to and by me will be sent outside of Blackboard using your student or personal email.

Your grade will be based on the following activities:

Activity	%	Point Value

Lab Assignments/Attendance	-2	10 points subtracted for each hour missed (2 sessions can be made up)
Class Lecture/Attendance	-2	10 points subtracted for each hour missed (2 sessions can be made up)
Homework	20	100 points total for all programs
Project I	20	100 points
Midterm	20	100 points
Project II	20	100 points
Final	20	100 points
Total Points	**500**	

Grading

Your grade will be calculated as follows:

Grade	%	No. of points	Grade	%	No. of points
A	90 and above	450 and above	D	60 - 69.9	300-349
B	80 - 89.9	400-449	F	Below 60	299 and below
C	70 - 79.9	350-399			

Tentative Class Schedule(16week normal semester session)

Weeks 1-8 -> Cover IDE's, primitive data types, file i/o up to language constructs. Basically 1) to 6) and 10) in the course outline.

Week 1 → Chapter 1, Install IDE

Week 2 → Chapter 2, Install Git, 10 Problems from Chapter 2 assigned.

Week 3 → Chapter 3, 10 Problems from Chapter 3 assigned, Chapter 2 problems due.

Week 4 → Chapter 3,4 10 Problems from Chapter 4 assigned, Chapter 3 problems due.

Week 5 → Chapter 4, Sign up for Gliffy Account

Week 6 → Chapter 5, 10 Problems from Chapter 5 assigned, Chapter 4 problems due.

Week 7 → Chapter 5, Chapter 5 problems due.

Week 8 → Chapter 6, Midterm, Project 1 assigned.

Weeks 9-16 -> More advanced concepts 7) to 11) in the course outline.

Week 9 → Chapter 6, 10 problems from Chapter 6 assigned.

Week 10→ Chapter 6, Project 1 due.

Week 11→ Chapter 7, 10 problems from Chapter 7 assigned, Chapter 6 problems due.

Week 12→ Chapter 7,8 10 Problems from Chapter 8 assigned, Chapter 7 problems due.

Week 13→ Chapter 8, Project 2 assigned, Chapter 8 problems due.

Week 14→ Chapter 9, Chapter 9 and 11 concepts assigned in lab.

Week 15→ Chapter 11, Project 2 due.

Week 16→ Chapter 13 Final and Project 2 class presentation.

Intersession(6week)

Weeks 1-3 -> Cover IDE's, primitive data types, file i/o up to language constructs. Basically 1) to 6) and 10) in the course outline.

Week 4 -> Midterm and Project I.

Weeks 4-5 -> More advanced concepts 7) to 11) in the course outline.

Week 6 -> Final and Project II.

Specific Tasks by Week for 6 week schedule.

Week 1 → Chapters 1 and 2 Gaddis 10 Programming Projects/Chapter

Week 2 → Chapters 3 and 4 Gaddis 10 Programming Projects/Chapter

Week 3 → Chapters 5 and 6 Gaddis 10 Programming Projects/Chapter

Week 4 → Midterm and Chapter 7 Gaddis 10 Programming Projects/Chapter

Week 5 → Project 1 and Chapter 8 Gaddis 10 Programming Projects/Chapter

Week 6 → Final and Project 2 Review of Chapters 9,11,13 from Gaddis

Lab Assignments will be assigned to support the Homework Assignments as needed.