IMPORTANT

The Final Exam will be held on Zoom on Wednesday, December 13th, 2023,

starting at 4 PM and ending at 6 PM.

Camera must be on during the whole exam.

Please make arrangements to make sure you are available that day between those times

AND that your computer camera and sound, as well as your Zoom app, are ready

CIS 121 Introduction to Programming with C++ (online)

FA 2023

Rafael J. Vicente

Office hours: **online**

E-mail: rafaelv@qcc.mass.edu
Extension: x3466

Text: C++ Programming: From Problem Analysis to Program Design, 8th ed, D. S. Malik, Cengage Learning

ISBN: 9781337684392

Description

This course focuses on the basic concept of programming, utilization of the executable codes, and implementation of these codes in problem solving. Students learn the concept of solving problems through the design and implementation of algorithmic solutions using the C++ programming language. Topics include the programming process, structured programming techniques, and basic logic formations. Practical business applications are emphasized throughout the course.

Prerequisite: Placement into college level English, MAT 095 with a grade of 'C' or higher or appropriate placement score Corequisite: CIS 111 or CIS 115 Semesters Offered: F/S

Objectives /Learning Outcomes

Students will:

- Learn the C++ environment and how to manage files, display output and compile programs. - Learn the fundamentals of the programming process through an understanding of logic, design, function and basic data structures.

- Learn about types, basic type conversion (casting)

- Learn how to use variables

- Learn how to use operators, increment, decrement and assignment operators to build more complex problems

- Learn logical and relational operators,

- Learn to write C++ code using selection constructs and loops.

- Learn how to use one-dimensional arrays

- Learn how to write functions, using parameters and returned values in the context of call by value and call by reference

- Learn the basics of input output (to the console and to files)

Online Liaison Service

The online liaison helps students succeed in their online courses. If online instructors are concerned that a student is having trouble using Blackboard or has fallen behind, they can contact the liaison to reach out and help that student. Online students also can contact the online liaison if they need assistance.

We are lucky to have **Lissa Walls**, who also teaches science classes at the College, as our online liaison. If you need help with or have questions about Blackboard Learn, please don't hesitate to contact Lissa. Below are several ways you can reach her.

- **EMAIL**: onlinecoursehelp@qcc.mass.edu
- **TEXT or CALL**: 774-285-3754
- **LOGON TO STARFISH**: Once you're in Starfish, you can find the online liaison in your success network and make an appointment with her using the "See available appointments" option.

Additional Assistance

If Lissa is not available and you need immediate assistance, you can do any of the following:

- **Call (508) 854-4427**. Follow the prompts for Blackboard Student Support for password or other technical issues.
- **Use Live Web Chat** to talk with a Blackboard specialist in real time by going to https://liveperson.edusupportcenter.com/sims/helpcenter/LiveChat.seam?inst_name=quinsigamond_cc.

If you have questions about course content or getting back on track with your coursework, I encourage you to contact me directly. Don't hesitate, though, to contact Lissa with your Blackboard Learn questions.

The General Academic Areas Tutoring Center

Room 222 Harrington Learning Center | 508-854-4279 | gaa@qcc.mass.edu

www.qcc.edu/gaa | Blackboard: On the Blackboard home page under “My Courses and Communities,” find “General Academic Areas Tutoring Center”

The General Academic Areas (GAA) Tutoring Center offers free, appointment-based academic assistance for a variety of subjects. At the GAA, you can work with a tutor to help clarify your course concepts and build your study skills. You can seek out a tutor at any point in the semester, for any question you may have about your course content, any time you feel like you need assistance.

All tutoring is by appointment. We offer both in person and online tutoring for all subjects. You can seek out tutoring individually (just you and the tutor) or bring up to five of your classmates for a small group session.

Tutors can also assist you with navigating Blackboard and Zoom, and accessing and navigating your online course products

To view our current schedule, visit our website above or scan the QR code below. Then email gaa@qcc.mass.edu to schedule an appointment with the day and time you’d like. We look forward to seeing you!

!

Accessibility Statement

Student Accessibility Services works to promote access to ensure an accessible college experience for students. Determining reasonable access and accommodations requires consideration of course design, course learning objectives and the individual academic and course barriers experienced by the student. If you have further questions, contact Student Accessibility Services. All discussions are confidential.

Quinsigamond Community College is committed to providing access and inclusion for all persons with disabilities. Students who require an accommodation in this course should notify the professor as soon as possible. Students are responsible for forwarding the Accommodation Letter to the professor (via email or hard copy). Students may request accommodations at any time during the semester, which begin upon receipt (accommodations are not retroactive). Please discuss any barriers which may arise during the semester with your Professor or Coordinator in the Student Accessibility Services office.Student Accessibility Services Contact Information:

Call: 508-854-4471

Email: disabilityservices@qcc.mass.edu

Services for Veterans

If you are a veteran of the armed forces, please visit the Veteran Affairs Office located in 258A (Administration Building) or contact them at veteranaffairs@qcc.mass.edu

Academic Honesty and Plagiarism

Our purpose in the classroom is to seek the truth; this work requires trust and honesty between teacher and student. If we are not honest about what we know and don't know, our learning will always be

impaired. Because our teaching and learning depends on this honest communication, we expect all students to understand what plagiarism is and why it is unacceptable.

Plagiarism means taking someone else's ideas or words and presenting them as one's own. The offense can take many forms including cheating on a test, passing in a paper taken from the Internet or from another student, or failing to properly use and credit sources in an essay.
Sometimes the issue is subtle, involving getting too much help on an assignment from someone else. In every instance, plagiarism means cheating both oneself and the owner of the source. Since cheating sabotages a student's learning experience, consequences range from no credit for the assignment to failure for the course and possible expulsion from the college.

Any student considering plagiarism should recognize the consequences and consider alternatives. Students uncertain about what constitutes plagiarism may request help from faculty or from appropriate college services.

Course Outline

The dates below are tentative

Topic	Source	Dates
The C++ environment. Binary numbers and internal representation. Thinking about programs: Top -down design, data structures, algorithms, functions. Compiling a Simple Program Comments Variables and Data Types	Notes and Chapter 1	**09/06 – 09/13** **Assignment 1:** **due 09/13**
Integer, char, floating point, booleans , string. Basic operators, precedence. Basic Input/Output Increment/Decrement operations Assignment and Casting. The result of the assignment operator, Chained assignments	Chapter 2, pages 28-84 and notes	09/14 – 09/22 **Assignment 2:** **Due 09/22**
I/O Streams More on Functions	Chapter 3, pages 123129, 133-135, 140143 and notes	09/23-09/30 **Assignment 3:** **due 09/30**

Control structures I : Branching with ifelse. Boolean values, relational and logical operators.	Chapter 4, pages 186-211 and Notes	09/30 – 10/7 **Assignment 4: due 10/07**
Functions, declarations, prototypes, local and global variables.	Chapter 6, pages 348-358 and Notes	10/07 – 10/15 **Assignment 4B: Due 10/15**
Control Structures II: Switch	Chapter 4, pages 227-234 and Notes	10/15 – 10/24 **Assignment 5: due 10/24**

Loops I: The need for looping; while statement	Chapter 5, pages 266-280 and notes	**10/24 – 11/02** **Assignment 6: due 11/02**

TEST 1		**11/6**
Functions 2: parameters and arguments. Passing by value and by reference. Static and automatic variables	Chapter 6, pages 384-411and Lesson 7 (notes)	11/7 – 11/14 **Assignment 7: due 11/14**
Loops II: For loops. Counters, nested loops. Translating for loops into while loops Do-while loops. Comparing loop statements. Break, continue.	Chapter 5, pages 298-305, 309-316 and notes	11/14 – 11/22 **Assignment 8: due 11/22**

Introduction to Arrays. Indexing, looping through selected elements of an array. Initialization. The meaning of the name on an array. C-Strings.	Chapter 8, pages 521=540 and notes	11/27 – 12/07 **Assignment 9: due 12/07**
FINAL	**Wednesday, December 13, 2023 From 4:00 PM to 6:00 PM , on Zoom.**	

Requirements

Homework: Students are expected to devote to their homework enough time to seriously attempt the solution of the programing assignments. Programming exercises should be submitted by the specified deadline. Programming assignments are individual tasks; students should make sure they can explain and/or clarify the code they have written if questioned by the instructor. Late submissions may result in a loss of 10 points per assignment / day.

No assignments will be accepted after the last assignment is due (in our case, after December 07, 2023).

Attendance: Students are expected to sign on to the online instruction site every day in oder to check for new announcements, assignments or readings.

*Tests/Main Projects***:** Students will take the test and the final at the scheduled time, which will be announced in advance. A student who does not take the test or the final as scheduled will receive zero points for that exam. Exceptional circumstances may be considered.

The FINAL EXAM will be held online using Zoom. Your computer camera and sound must be active during the whole final exam, which will be two hours long. The whole class must take the final on the same day and at the same time. The date and time are specified in the Course Outline section of this syllabus. Please make sure you have no other commitments on that date and time.

Textbook: Required.

Compiler/IDE: A compiler/IDE is required. A recommended (free) compiler is suggested in the Announcements section of Blackboard.

Readings: Any supplementary readings, as posted or recommended on the web site and/or assignments, are required and may be included on tests and subsequent assignments.

- *Academic honesty*: programming requires individual effort and accountability. Any assignment or test submitted must be the result of the student's personal effort. Students must be able to explain unequivocally any component of their work.

  Including someone else's work in an assignment or test is a very serious violation of the course rules and may result in a failing grade.

Interaction

Students will **interact** with the instructor via QCC **email** and using Zoom (appointment required).

Students can make an appointment at any time to meet electronically with the instructor in order to discuss any course-related topics.

In addition, students are encouraged to take advantage of the instructor's online availability, An appointment is needed to prevent schedule conflicts with other students.

Evaluation procedure and Grading Policy

The final grade will be computed as follows:

Average of Assignments: *70 %*

Two tests or Projects : *30 % (15% each)*

The final grade will follow the QCC letter grade conventions.

A 95 – 100	B – 80 – 82	D + 67 – 69
A– 90 – 94	C + 77 – 79	D 63 – 66
B+ 87 – 89	C 73 – 76	D – 60 – 62
B 83 – 86	C – 70 – 72	F 0 – 59

Assignments: the programming exercises are intended to illustrate the theory and to set the basis for more advanced topics. A correct programming exercise should compile and run as requested. If a submitted programming exercise fails to run as required, the student may be granted a brief extension (two days) to review it, but s/he will have to interact with the instructor during the process of refining the original submission.

Partial credit may be considered if a project that does not run as required.

Most assignments include a quiz component in addition to the programming exercise. Assignments will be graded as follows:

Quiz Component : 30% (if any)

Programming component : 70%

The quiz component (theory questions) typically requires short answers. The following grading criteria will be applied:

90-100 : right answer, maybe with some inconsequential error

70-89: some of the fundamentals are right, but the answer is not functionally right (i.e. would not work in a program)

50-69 : some serious errors, but at least half right

0-49 : serious errors, little or no correct content

The grades on the programming component of an assignment will range between:

90-100: the program runs perfectly on any input and the program is built according to the required specications.

80-89 : the program runs well for most inputs, but fails under limited conditions. Program design specs are met.

70-79 : the program runs well for common inut, but fails to take into account special cases. Program design specs are met

50-69 : the program fails to run due to subtle errors or the program design specs are not met
049: the program does not run for any input, does not compile or solves a different problem than the one requested

Note on Plagiarism: Programming requires individual effort and accountability. Plagiarism is a very serious violation of the course rules and may result in a failing grade. All code include in the programming assignments must be created by the student, who must be able to explain it in detail in order to be considered the author of the program.

Methodology and Teaching Procedures

The course will be delivered online. Interaction among students and student-instructor interaction will be provided via e-mail, and virtual rooms using Blacboard's Virtual Meeting or Skype, at instructor's discretion.

Assignments on the major course topics will be posted on average, approximately, every ten days. Assignments are to be completed by their deadline, but an extension on the programming

component maybe granted if the student has made enough progress towards its completion, and the instructor deems such an extension educationally useful.

In addition to the assignments, the instructor may ask the class to comment/interact on a specific topic via e-mail or through a discussion group specific to that topic.

Students are encouraged to inquire and seek clarification about the programming assignments and/or the concepts those assignments are based upon.

In general, assignments will be structured as follows:

-***Readings*** *from the textbook, Blackboard and/or Web sites*

- ***Questions*** *to be answered based on the assigned readings*

-Program*: programming component*

The following study **workflow** is recommended:

1. Identify the current assignment
2. Read the introductory materials and the relevant textbook section(s), as specified in the assignment
3. Contact the instructor with any questions
4. Try to run some of the examples included in the readings. Experiment with the code by introducing your own variations
5. Ask any questions raised by the previous step
6. Start the assigned exercises (quiz component and program) 7. 7. Ask for clarification, if needed

Tentative Test/assignment Schedule

The tentative test schedule is included in the Course Outline table included above.

Assignments will be posted approximately every seven to ten days.

Assignments are expected to be submitted by their due date. Extensions may be granted if the submitted assignment is near completion and the instructor deems that a few corrections might lead to a satisfactory solution.