# CSC 211 - Computer Programming

## Course Instructors

**Instructor:** Michael Conti (michaelconti@uri.edu), Tyler 132

**Lecture:** Tue/Thurs | 12:30p - 1:45p | East Hall Aud

**Teaching Assistants:**

- **Undergraduate:** Genevieve, Marvens, Jenny

**Office Hours Schedule**

| Day | Staff Member | Time | Location |
|---|---|---|---|
| Monday | Marvens<br>Genevieve | 12p - 1p<br>4p - 5p | Tyler 3rd Floor<br>Tyler 3rd Floor |
| Tuesday | Michael<br>Genevieve<br>Marvens | 10a - 11a<br>2p - 5p<br>4p - 6p | Tyler 132<br>Tyler 3rd Floor<br>Online |
| Wednesday | Marvens<br>Genevieve<br>Genevieve | 12:30p - 2:30p<br>1:30p - 2:30p<br>5p - 6p | Tyler 3rd Floor<br>Tyler 3rd Floor<br>Online |
| Thursday | Marvens<br>Genevieve | 10a - 12p<br>2p - 5p | Online<br>Tyler 3rd Floor |
| Friday | Genevieve | 4p - 5p | Tyler 3rd Floor |

**Discussion Session**

> Attending 80% of discussion sessions will earn you an additional 5pts on your final exam.

| Day | Staff Member | Time | Location |
|---|---|---|---|
| Monday | Marvens | 1p - 2p | Tyler 3rd Floor |
| Thursday | Genevieve | 11a - 12p | Tyler 3rd Floor |

**Lab**

| Day | Staff Member | Time | Room |
|---|---|---|---|
| Monday | Genevieve<br>Jenny | 2:00p - 3:45p | Tyler 055 |
| Friday | Marvens | 12:00p - 1:45p | Library 166 |
| Friday | Genevieve<br>Jenny | 2:30p - 3:45p | Library 166 |

## Course Overview

CSC 211 provides a rigorous introduction to computer programming using the C/C++ language and object orientation. The course also explores basic computational problem-solving techniques, algorithms, and elementary data structures. Prior programming experience is not strictly necessary, however, students must be familiar with the basics of computers.

Prerequisites: CSC 106 or major in Computer Engineering.

## Student Learning Outcomes

Upon successful completion of this course, each student will be able to:

- Describe how data are represented in memory (stack/heap)
- Write programs of moderate complexity in C++
- Implement solutions that involve recursive functions
- Implement and use elementary data structures, including arrays/vectors and linked lists
- Reason about the computational costs of certain basic operations
- Decompose problems and develop abstractions to simplify problem solving
- Write programs using OOP concepts (e.g., objects, classes, encapsulation, polymorphism, and inheritance)

## Required Textbooks

- Problem Solving with C++, 10th Ed., W. Savitch.

## Lab Sessions

Lab sessions will be held on Thursday and Mondays. Lab sessions involve short explanations given by TAs or the instructor, followed by a set of programming exercises. Students are required to solve all programming exercises during the lab session and turn in solutions through Gradescope. Lab work is not graded work, however, attendance will be part of the final grade. We use your submitted solutions to record your attendance.

## Programming Assignments

Programming assignments are individual work. Students will have roughly 7 days to work on each assignment, and there will be approximately 8 assignments in total. Each programming assignment has a specific due date/time listed on the course web site. Late submissions will not be accepted. All programming assignments are automatically graded on Gradescope. For each of the questions you either pass the test cases (full points awarded) or not (zero points). Partial credit on individual questions is not awarded. Students are strongly encouraged to bring their code to TA or instructor's office hours prior to the due date.

## IDE Selection

You are free to use any IDE for developing your programming assignments and working on the lab sessions. However, the source code you submit for programming assignments must compile without any errors on a linux station and a g++ compiler. We strongly recommend using CS50 IDE. You can decide to install the offline CS50 IDE on your computer as a containerized app, for which, installing docker is necessary.

Alternatively you can use CS50 IDE online, for which you only need to have a free GitHub account. You can also refer to the CS50 IDE FAQs if you want to know more about the IDE.

## Exams

Exams are closed-book and held during lecture times. You are allowed to bring a cheat sheet to every exam. This reference page is a single sheet, in which you can include hand-written annotations only, on both sides. Students will be notified of the contents prior to the exam. Make-up exams are given only in rare cases of documented events.

## Discussion Sections

Depending on class demand, students can choose to participate in one discussion section per week where we review topics lightly covered in lectures introduce concepts not covered in class but useful for assignments, and provide exam reviews.

## Grading

Coursework consists of lab attendance, problem sets, programming assignments, and exams. Your final grade will be calculated according to the following table:

| Gradebook item | Count | Weight |
|---|---|---|
| Lab Attendance | 1 | 10% |
| Programming Assignments | ~5 | 25% |
| Weekly Programming Challenges | ~6 | 10% |
| Exams | 2 | 30% |
| Final Exam | 1 | 25% |

Your final letter grade will be calculated using the cutoffs in the table below. These cutoffs might be lowered, but they will not be raised. Your final letter grade will be the letter corresponding to the highest cutoff value less or equal than your final grade. Consider that those values are strict. For example, a final grade of 93.99 is an A- and not an A.

| | | | | | |
|---|---|---|---|---|---|
| **A [94-100]** | **A- [90-92]** | **B+ [87-89]** | **B [83-86]** | **B- [80-82]** | **C+ [77-79]** |
| **C [73-76]** | **C- [70-72]** | **D+ [67-69]** | **D [60-66]** | **F [< 60]** | |

## Academic Enhancement Center

Located in Roosevelt Hall, the AEC offers free face-to-face and web-based services to undergraduate students seeking academic support. Peer tutoring is available for STEM-related courses by appointment online and in-person. The Writing Center offers peer tutoring focused on supporting undergraduate writers at any stage of a writing assignment. The UCS160 course and academic skills consultations offer students strategies and activities aimed at improving their studying and test-taking skills. Complete details about each of these programs, up-to-date schedules, contact information and self-service study resources are all available on the AEC website, uri.edu/aec.

## Academic Honesty

Discussions with others to understand general homework problems and class-related concepts are strongly encouraged. However, when working on assignments, all written work and source code must be your own. You might not look at anyone's written solution. Students are prohibited from accessing or comparing homework answers with those of other students prior to submitting each assignment. Copying another individual solution is plagiarism, a serious offense, and the one most common in computer science courses. Anyone that provides homework answers, program code for a programming assignment to another individual is also guilty of academic dishonesty. Both will be prosecuted in accordance with the University's Policy of Academic Honesty. If you do not have sufficient time to complete an assignment, then submit a partial solution.

## Anti-Bias Syllabus Statement

We respect the rights and dignity of each individual and group. We reject prejudice and intolerance, and we work to understand differences. We believe that equity and inclusion are critical components for campus community members to thrive. If you are a target or a witness of a bias incident, you are encouraged to submit a report to the URI Bias Response Team at www.uri.edu/brt. There you will also find people and resources to help.

## Disability, Access, and Inclusion Services for Students Statement:

Your access in this course is important. Please send me your Disability, Access, and Inclusion (DAI) accommodation letter early in the semester so that we have adequate time to discuss and arrange your approved academic accommodations. If you have not yet established services through DAI, please contact them to engage in a confidential conversation about the process for requesting reasonable accommodations in the classroom. DAI can be reached by calling: 401-874-2098, visiting: web.uri.edu/disability, or emailing: dai@etal.uri.edu.

## Religious Holidays

It is the policy of the University of Rhode Island to accord students, on an individual basis, the opportunity to observe their traditional religious holidays. Students desiring to observe a holiday of special importance must provide written notification to each instructor.

## Tentative Course Outline:

> The weekly schedule is subject to change per instructor discretion.

| Week | Topics | Resources |
|---|---|---|
| Week 1 | Lecture - Introduction to 211, Computer Systems, Programming Languages<br>Lab - Hello 211, IDE Setup, Basic Shell Commands<br>Reading - Savitch, Chapter 1 | Lecture Slides<br>Lab |
| Week 2 | Lecture - Problems/Algorithms/Programs, History of C++, The Compiler<br>Lecture - C++ Basics, Input/Output, Data Types, Expressions<br>Lab - Algorithms, Problem Design, Pseudo-code Exercises<br>Reading - Savitch, Chapter 2 | Lecture Slides<br>Lecture Slides<br>Lab<br>Assignment00 |

| Week | Topics | Resources |
|---|---|---|
| Week 3 | Lecture - Number Systems, Further look into DataTypes<br>Lecture - Expressions, Selection Statements<br>Assignment - Assignment 0<br>Lab - Programming Exercises (branching)<br>Reading- Savitch, Chapter 3 | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 4 | Lecture - Introduction to Loops (for)<br>Lecture - Loops (while, do while) and Nested Loops (examples)<br>Assignment - Assignment 1<br>Lab - Programming Exercises (loops and nested loops)<br>Reading- Savitch, Chapter 3 | Lecture Slides<br>Lecture Slides<br>Lab<br>Assignment01 |
| Week 5 | Lecture - Functions<br>Lecture -Scope of Variables, Parameter passing, Call Stack<br>Lab - Using the Debugger, Programming Exercises (functions)<br>Reading - Savitch, Chapter 4<br>Reading - Savitch, Chapter 5 | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 6 | Lecture - Arrays, Arrays and Functions<br>**Exam - Midterm Exam (weeks 1 to 5)**<br>Lab - Strings (C style strings and string objects)<br>Assignment - Assignment 2<br>Reading - Savitch, Chapter 7<br>Reading - Savitch, Chapter 8 | Lecture Slides<br>Lecture Slides<br>Lab<br>Assignment02 |
| Week 7 | Lecture - Basic Sorting<br>Lab - Basic sorting algorithms | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 8 | Lecture - Multidimensional Arrays<br>Lecture - Pointers<br>Lab - Programming Exercises (pointers)<br>Reading - Savitch, Chapter 9 | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 9 | Assignment - Assignment 3<br>Lecture - Recursion and Examples<br>Lecture - Recursion (cont.) and Examples<br>Lab - Programming Exercises (tracing recursion, drawing recursion trees) | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 10 | Lecture - Binary Search<br>Lecture - Advanced Recursion (Backtracking), Structs<br>Lab - Advanced Recursive Problems | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 11 | Assignment - Assignment 4<br>Lecture - Classes, Data Members and Methods (Encapsulation)<br>**Exam - Midterm Exam (weeks 6 to 10)**<br>Lab - Implementing Classes (source/headers), Arrays and Objects | Lecture Slides<br>Lecture Slides<br>Lab |

| Week | Topics | Resources |
| --- | --- | --- |
| Week 12 | Lecture - Constructors<br>Lecture - Dynamic Memory Allocation, Destructors<br>Lab - Developing a string Class (overloaded operators and copy constructors)<br>Reading - Savitch, Chapter 14 | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 13 | Lecture - Class Inheritance<br>Lecture - Singly Linked Lists<br>Lab - STL Containers, read/write from files, and CLAs<br>Reading - Savitch, Chapter 15 | Lecture Slides<br>Lecture Slides<br>Lab |
| Week 14 | **Exam - Final Exam (cumulative with focus on weeks 11 to 14)** | None |