

# CPSC 122 Computer Science I

## Syllabus Fall 2021

[Gonzaga University](https://www.gonzaga.edu)

*(Note: syllabus subject to change, your instructor will make an announcement if changes occur – Last updated: 8/28/2021)*

---

### Instructor Information

Instructor: [Daniel Olivares](#), PhD

Office: Herak 309A

Office phone number: 509-313-5753

Email: [olivares@gonzaga.edu](mailto:olivares@gonzaga.edu) *(All course-related messages should be sent via **Canvas** when possible)*

Office Hours:

Monday and Wednesday: 2:05PM-4:05PM **and by appointment**

### Course Information

- A continuation of CPSC 121. An examination of pointers and recursion; an introduction to basic data structures and algorithmic analysis.
- Credits: 3.00
- College: School of Engineering/Applied Science (SEAS)
- Department: Computer Science
- Prerequisites: CPSC 121

### Description

CPSC 122 is a continuation of the first course in computer science (CPSC 121) for majors. In this course, we use the C++ programming language to continue to explore the fundamental concepts, constructs, and techniques of modern computer programming, including (but not limited to) sorting and searching algorithms, pointers and dynamic memory management, classes and data types, and recursion. The primary aim of this course is to give you a thorough introduction into problem solving, algorithm discovery, and program design in C++.

### Course Times and Location

- Section 01: 10:00AM - 10:50AM; Tuesday, Herak 223
- Section 02: 11:00AM - 11:50AM; Tuesday, Herak 223

### What we will Learn

Students who successfully complete this course will be able to:

1. Design and develop software using basic object-oriented programming techniques,
2. Define and implement basic linear data structures, including linked lists, stacks, and queues Analyze algorithms with stacks, queues, and lists,
3. Design and develop recursive solutions.

Outcomes:

- Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

- Apply computer science theory and software development fundamentals to produce computing-based solutions.

## Course Materials

### Course Website

Canvas LMS: <https://canvas.instructure.com/>

**Note:** ALL course materials will be found on the Canvas LMS.

**Note2:** Please use the indicated URL, do not Google search for Canvas and use the first link in the results! This can lead to trying to log into the wrong canvas system and the false belief that you are locked out of your account.

## Schedule

For an up-to-date and detailed schedule, please see the downloadable version available on the course website.

### Course Topics

- Algorithms and analysis
  - Basic sorting algorithms (bubble sort and selection sort)
  - Basic searching algorithms (linear search, binary search)
  - Algorithm efficiency and informal introduction to Big O notation
- Programming in C++
  - File I/O
  - Pointers
- Basic object-oriented programming in C++
  - Definition of classes including data members, member functions, constructors, destructors, copy constructors, and copy assignment operators
  - Public vs. private data members and functions
  - Header files and multi-file compilation
  - Basic reuse mechanisms
    - Composition (current class uses a pointer to a class as a member variable)
    - Simple inheritance (current class is a public subclass of another class, including virtual and pure virtual functions)
- Dynamic memory management in C++
  - New and delete (primitives, arrays, objects)
  - Single, circular, and doubly linked lists
- Basic abstract data types and their implementation
  - Lists
  - Stacks
  - Queues (conventional, priority, circular arrays)
- Recursion
  - Recursively defined mathematical functions (e.g., exponentiation, factorial)
  - Recursively defined string functions (length, display contents forward and in reverse)
  - Classic recursive functions (Fibonacci and Towers of Hanoi or N-Queens)
  - Recursive traversal of lists Recursive implementation of binary search

## Communication

We will use Canvas to communicate, submit assignments, and view grades. A URL invitation link will be sent to your official @zagmail.gonzaga.edu email to provide course access.

**Note:** Please use Canvas as the primary communication method for course-related messages. I will monitor email as well but using Canvas is the preferred communication method. This will

increase your message visibility and reduce likelihood of emails getting flagged as spam or getting lost in transit. Further, **any course-related emails should be sent from your official @zagmail.gonzaga.edu student email and include “CPSC 122” in the subject line.**

Additionally, [Discord](#) (free to use) will be used to augment class communication and facilitate digital office hours—ask questions and discuss topics with other students in the class, TAs, and the instructor. Discord supports voice and text communication as well as screen sharing capabilities (see *Canvas for server invite URL*).

**Zoom** will also be used to augment course interactions as necessary. **Please log in with your Gonzaga credentials and not separate free Zoom account credentials** (use the SSO log in method with Gonzaga as the domain).

**All communication methods are not to be used to share code solutions** (see *academic honesty policy*). You can, however, post high level code explanations and/or snippets of pseudocode. I will also post/email important information to you through Canvas and Discord announcements channels/feed. **You are expected to check announcements on Canvas and your GU email regularly.**

## Course (and Digital) Classroom Etiquette

- Please respect the food and drinks policy in the classroom and use common sense (i.e., don't damage lab equipment!)
- Please be conscious of appropriate behavior and background while communicating via digital modes.

## Textbooks

### Required:

- *Starting out with C++: From Control Structures through Objects* by Tony Gaddis. 9th Edition.
  - i. Print ISBN: 9780134498379
  - ii. eText ISBN: 9780134443850

The required text book can be found in various formats. If you prefer, you can purchase the paper version. For digital versions, you have a few options, e.g., a 180 day rental will be cheaper and if you choose to purchase through VitalSource there is a “lifetime” purchase option that will not expire once downloaded.

## Required Hardware

A laptop adhering to GU SEAS requirements. Click [here](#) to learn more about the requirements.

## Required Software

- *Preferred development environment:*
  - i. Windows 10 with Visual Studio Community.
- *Alternate development environments:*
  - i. Linux or MacOS (or Windows 10) with Visual Studio Code configured with g++ and the visual debugging extensions.
  - ii. [VirtualBox](#), which is available for Windows, MacOS, and Linux if you want to use the CS department Linux image as your development environment. This image includes the g++ compiler, gedit, and/or Visual Studio Code.
  - iii. GU-Hosted Linux image. This is the same image used in alternative ii but is accessed via a browser window.

- iv. CS50IDE. This is an externally hosted (non GU affiliated) development environment providing a simple text editor and command line compiler. Like option iii, it is also accessed via a browser window.
- Guides will be available on the Canvas course home page for each of these methods (excluding MacOS, sorry!).

## Course Environment

This is an *active learning class*. You are expected to come to class prepared, actively attend and participate in class, and to participate regularly in discussions on Canvas outside of class (*Discord involvement is not required though will provide additional opportunities to communicate with your peers and seek help*). In class, we will be working several coding/computing tasks and **it is expected that you will bring your laptop and actively participate.**

**Canvas** is the online presence for this course. You can access it at <https://canvas.instructure.com/>. Once you log on to our course site, you can read course announcements, participate in online discussions, send messages to course participants, access course materials, hand in course deliverables, review peers' work, and access your grades. This is where you will also find all private URLs, e.g., Zoom, and Discord invites.

## Food & Drink Policy

Please respect the specific classroom food/drink policy, e.g., rooms with computer workstations will not allow food or drink.

## Course Grading

### Assignment Weights

- Participation Activities (7%)
- Quizzes (8%)
- Programming assignments (50%)
- Exams (35%)
  - Exam 1 (7%)
  - Exam 2 (12%)
  - Lab final exam (16%)

## Assignment Categories

### Participation Activities (7%)

Class participation is expected and is a vital part to successful completion of this course. I understand that you may need to miss class occasionally for valid reasons. For this reason, your **three lowest participation activity scores will be dropped**—that is, you will receive three free attendance/participation credits. Any discrepancies in participation need to be brought to my attention within a week of the posted grade.

Participation activities are credit/no credit and will be scored based on submission effort. I understand that sometimes there are difficulties understanding/completing participation tasks. Submissions that display **minimal/no effort will not receive credit!** Make an honest effort to complete the given tasks for participation credit. **For any incomplete participation you must comment on your submission with 1) a description of what you are struggling with, 2) what you tried that didn't work, and 3) which specific resources (e.g., that day's lecture slides, book chapter(s), etc.) you used to attempt to understand/solve the participation activity.**

### Quizzes (8%)

For frequent practice with memory-retrieval and problem solving, there are regular quizzes. The quizzes are individual quizzes: You come up with the solution to the problem on your own and submit your solution to the problem individually.

Note: *I will drop your 2 lowest quiz scores.* This means that you are given 2 quiz "freebies" that excuse your failure to submit a quiz for any reason.

### Programming Assignments (50%)

You will be given six (6) programming assignments (PAs) to complete. All C++ code written in assignments must adhere to the recommended CPSC 121 C++ Style and Coding Standards (see Canvas files for this document). Please upload assignments as directed on each assignment to the corresponding assignment in Canvas. *See the late work section for the PA late submission policy.*

Each PA will be worth 10% of your final grade each in the final grade calculations. The Grade weight for PAs will consist of 30% of your final grade for PA1-4 and 20% of your final grade for PA 5-6. Your lowest one (1) PA score from PAs 1-4 will be dropped from grade calculation for that 30% of your final grade. For example, if you score 95%, 70%, 100%, 100% for PAs 1-4, the 70% score will not be used to calculate your final grade resulting in a ~2% higher grade.

### Exams (35%)

We will have two exams and one lab final exam in this course. **Please see the course calendar for your midterm and final exam dates.** The official finals period is:

- Section 01: Tuesday, December 14..... 1:00 pm to 3:00 pm
- Section 02: Thursday, December 16.....8:00 am to 10:00 am

Exams may be rescheduled for students that have **valid excuses**. To increase your chances of your excuse being determined "valid", notify the instructor no less than two days in advance if you are going to miss an exam.

*Note: Final exams may not be taken early. You must take your final exam at the time listed above for the course section you are enrolled in.*

### Grading Scale

In this course, your grade will be tracked as a percent, which will be mapped to a letter grade.

Please see the table for the conversion.

93-100%	A	73-76.99%	C
90-92.99%	A-	70-72.99%	C-
87-89.99%	B+	67-69.99%	D+
83-86.99%	B	60-66.99%	D
80-82.99%	B-	0-59.99%	F
77-79.99%	C+		

### Contesting a Grade

If you believe that a mistake has been made with grading an assignment or exam, *please speak with me **within one week (but no sooner than 24 hours) of the assignment or exam being returned.*** Do not wait until the end of the semester to discuss any grade changes. You need to constantly be aware of how you are performing in the class. Thus, there will not be any

surprises at the end of the semester when grades are to be formally submitted. You should be able to view your grades via Canvas. These will be updated regularly.

Note: the grades are **weighted** according to the ones described above. You cannot determine your grade with a raw point total.

## Suggestions for Getting the Most out of This Course

- **Attend class.** You can only benefit from this course if you show up! This is especially true of the group activities. I expect you to participate actively in class by asking questions, answering questions, and engaging in the collaborative design and problem-solving activities. Remember, part of your grade is based on attendance and participation (see above).
- **Put in enough time.** My rule of thumb is that students need to put in 3-4 hours of work outside of class for every hour they spend in class. This translates to roughly **6-8 hours per week**. You may need to put in only a fraction of 6 hours during some weeks, while you will find yourself putting in more than 6 hours during other weeks—especially during weeks in which your programming assignments are due.
- **Take initiative to get help.** You are strongly urged to make use of the many resources available to you! You can do this in two ways. First, I recommend that you find students in the course with whom to meet and discuss course material. Second, take the initiative to contact the instructor or other students if you begin to struggle. Do so sooner rather than later to minimize unproductive time spent working on assignments.
- **Have reasonable expectations.** Learning does not come "for free"; it is not simply a matter of "being taught." You'll get the most out of this course if you take an active role in your own learning and try to have fun doing so.

## Office Hours

You are strongly encouraged to take advantage of office hours **and/or make an appointment** to meet with me if you have questions about the course material. I am more than happy to help you and office hours are a great way to get one-on-one help with the material. You are not "bothering" your instructor if you show up to office hours! Those hours are set aside explicitly for you to use.

As an alternative to face-to-face office hours, I will also be holding digital office hours via the class Discord server which provides text and voice communication augmented with screen sharing.

**Note:** Even though programming assignments are noted as individual assignments, this does NOT mean that you are not allowed to get help with the assignments. You are encouraged to make use of help from your instructor, TAs, or tutoring services provided the work you submit is your own (that is, someone else did not produce the work you are submitting).

## Course Policies

Please familiarize yourself with the following course policies. By following them, you will get the most out of this course, and you will not encounter any unwelcome surprises down the road.

- **Add a profile picture to Canvas and Zoom.** Uploading a recent picture of yourself to Canvas will help me, the TA, and other students in the class to associate your name with your face. I would greatly appreciate it if you would do this, as it will help me to learn your name more quickly.
- **Corresponding with the instructor via e-mail.** Please e-mail me only through Canvas; do not e-mail me directly, except in an emergency. If you think your question is of general interest to the class, consider posting it to the course activity feed in Canvas. In general, you can expect an e-mail response from me quickly, and certainly within **24 hours** of sending it during the work week (excluding weekends and holidays). You must include “CPSC 122” in the subject line of any emails you send me outside of Canvas.
- **Accessing course materials.** Canvas is the online presence for this course. **Log in regularly (every day)** to view course announcements, view the course calendar and schedule, participate in the course feed, access course materials, access your grades, and submit assignments.
- **Checking your grades.** To view your current grades, click on the Grades tab in Canvas. My goal is to have work graded within one week of the final deadline, but this may not always be possible. Please check your grades regularly to ensure that your grades have been entered properly, and please let your instructor or the TA know as soon as possible if you detect an error.
- **Late policy for assignments.** **Deadline reminders are a courtesy, not a requirement. You are responsible to follow the course calendar and be aware of provided due dates!** Course assignments are due by the stated due dates and times. Please see each assignment prompt for that assignment’s late submission policy. **Note that some assignments may be time sensitive and will not allow for late submissions.** In cases of illness and extenuating personal circumstances, you may request via email that an exception be granted to this policy, but your request must be issued in a timely manner (preferably in advance of the due date), and there is no guarantee that it will be granted.
- **Programming Assignments (PAs)** may be turned in up to two days late, at a penalty of 10% per 24 hours late. Forty-eight hours after the assignment is due, you may *no longer hand in the PA for credit*.

#### **Submission Tips:**

- Do not plan to submit your assignment at the last minute! Submit early and often. You are allowed multiple submissions and are not penalized for submitting more than once. Your **latest** submission will be graded unless otherwise noted.
- Even if you fail to submit an assignment, I encourage you to work through the assignment and seek help as needed in order to ensure you understand the material completely.
- If an emergency occurs, I will accommodate the student as much as possible. Make-up exams will not be possible unless the student speaks with me at least two days in



advance. Emergencies do occur and rescheduling of exams because of these is up to my discretion.

## **Academic Integrity Policy**

### **University Policy**

You are expected to follow the university policy on academic honesty. Academic honesty is expected of all Gonzaga University students. Academic dishonesty includes, but is not limited to cheating, plagiarism, and theft. Any student found guilty of academic dishonesty is subject to disciplinary action, which may include, but is not limited to, (1) a failing grade for the test or assignment in question, (2) a failing grade for the course, or (3) a recommendation for dismissal from the University. A complete copy of Gonzaga's Academic Honesty policy can be found at [course catalog](#).

### **CPSC 122 Policy**

For this course both collaborative and individual work will be required.

- You are **RESPONSIBLE** for knowing all material involved in a collaborative assignment.

All individual work must be completed alone.

Do NOT work with any team members on individual assignments. You may **discuss ideas** with team members about problems related to individual assignments, but do not discuss implementation details.

Discussing implementation details includes (**but is not limited to**):

1. Copying/taking a picture of another student's code/work
2. Letting another student copy/take a picture of your code/work
3. Sending your code/work to another student (i.e. digitally or in print)
4. Receiving another's student code/work (i.e. digitally or in print)

Note: If you use content from sources other than the ones provided by the instructor (e.g. textbook, notes, etc.), **cite the source in your code**.

If are unsure of whether a situation might be considered cheating, be cautious and don't do it. If help is required, please ask the instructor for guidance. I'm always more than willing to help! Any instances of plagiarism will be reported to the Academic Integrity Board.

**Code plagiarism software will be used to check for code similarity.**



## Gonzaga University-Wide Policies

University Academic Policy Statements	Associated Links
Support Links	<a href="#"><u>Blackboard Link to Support Links</u></a>
Diversity, Equity and Inclusion	<a href="#"><u>Office of Diversity, Equity and Inclusion Bias Incident Assessment and Support (BIAS) Team</u></a>
Harassment, Discrimination and Sexual Misconduct Policies	<a href="#"><u>Harassment and Non-Discrimination Policy Title IX</u></a>
Academic Integrity Policy	<a href="#"><u>Academic Integrity Policy</u></a>
Students with Disabilities/Medical Conditions and accessible Documents (EITA)	<a href="#"><u>Disability Access and Resources Office Electronic Information Technology Accessibility (EITA)</u></a>
Religious Accommodations for Students	<a href="#"><u>Religious Accommodations for Students Policy</u></a>
FERPA and Privacy	<a href="#"><u>FERPA</u></a>
Class Attendance Policy	<a href="#"><u>Class Attendance Policy</u></a>
Notice to Students of COVID-19 Expectations	<a href="#"><u>Student Arrival &amp; Return to Gonzaga Guides</u></a>
Notice to Students about Class Recordings (audio, video and photos)	<a href="#"><u>Zoom sessions might be recorded</u></a>
Student Conduct	<a href="#"><u>Gonzaga University's Student Code of Conduct</u></a>
Course Evaluations	<a href="#"><u>Course Evaluations</u></a>