CPTS 223: Advanced Data Structures C/C++

Instructor: Yan Yan

Fall, 2024

Class Hours: M, W, F, 1:10PM - 2:00PM — Class Room: ABEL 201

Office Hours: Tue/Thu, 12:30PM - 1:30PM — Office: EME 123

TA:
Chibuike Emmanuel Ugwu
(Github username: chibuikeugwu)
Nathan Balcarcel
(Github username: nbalcarc)
Facundo Herrera Rivarola
(Github username: herrera-facundo)
Samuel Hickman
(Github username: shickmanIV)

TA office hour
- Chibuike Emmanuel Ugwu
  - 3:30PM-4:30PM on Tue/Wed — TA office: Dana 122
- Nathan Balcarcel
  - 3:30PM-4:30PM on Mon/Fri — EECS TA offices (Sloan 340-345)
- Facundo Herrera Rivarola
  - 2:30PM-3:30PM on Mon/Fri — EECS TA offices (Sloan 340-345)
- Samuel Hickman
  - 3:30PM-4:30PM on Wed, 4:30PM-5:30PM on Thu — EECS TA offices (Sloan 340-345)

Preferred Contact: **Canvas Message**
Alternative: Email
- Instructor: yan.yan1@wsu.edu
- TAs:
  - chibuike.ugwu@wsu.edu
  - f.herrerarivarola@wsu.edu
  - samuel.hickman@wsu.edu

---

1 Course Description

CptS 223 [1] is a computer science course for majors. In this course, we use the C++ programming language to design, develop, implement and analyze data structures and algorithms beyond the

[1] Some of these policies and descriptions were derived directly from Andrea O'Fallon's, Aaron Crandall's, Ananth Kalyanaraman's and Jia Yu's CptS 223 syllabi.

basic data structures discussed in CptS 122. This course also introduces you to algorithm design through different data structures and algorithmic techniques, and algorithm analysis using asymptotic notation and complexity analysis. One important component of this course is to have you design multiple algorithmic solutions to a single problem using different combinations of data structures that you learn in class, and then comparatively evaluate their relative efficiencies. Another important component of this class is to have you implement the data structures and algorithms using C++ and STL (Standard Template Library), and evaluate the algorithms empirically. You will also learn basic concepts in parallel and multithreaded programming such as how to analyze parallel runtime complexity, compute speedup; and synchronization primitives such as locking. You will be introduced to OpenMP multithreaded programming for shared memory multicore systems. You will have at least one programming project that expects you to come up with a parallel implementation for one of the data structures – e.g., parallel hash table construction and access. You will also learn to include parallel performance statistics (such as speedup) in results. This course will introduce gcc/CMake on the Linux platform as the development environment. Tools such as Git will be introduced for tracking source versions and will be the primary vehicle for turning in assignments. All projects will be tested on Ubuntu Desktop 20.04/22.04 (LTS) or macOS.

2 Learning Objectives and Expected Outcomes

At the conclusion of this course, you should be able to:

- Design, implement, and test programs by applying modern tools and techniques.
- Analyze and compare a variety of data structures including: BST, AVL, B, B+, splay, and red-black trees, hash tables, sets, heaps, and graphs.
- Design efficient algorithms.
- Setup and apply a version control system.
- Analyze a specification of a problem of moderate complexity and construct a structured, elegant C++ program that solves the problem in the Linux OS environment.
- Utilize basic parallel programming concepts along with OpenMP multithreaded programming for shared memory multicore systems.
- Identify and implement test cases to edge scenarios in pseudocode and/or C++ code.
- Identify, analyze, and solve C++ code interview questions in prep for internships.

The following are WSU and ABET outcomes. After completing this course all students will have an ability to:

- Analyze complex computing problems 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 a program's discipline.

- Communicate effectively in a variety of professional contexts.
- Apply computer science theory and software development fundamentals to produce computing-based solutions.
- Acquire and apply new knowledge as needed, using appropriate learning strategies.

3 Prerequisites

Before taking this course you need to satisfy the prerequisites (CPT S 122 with a C or better; MATH 216 with a C or better or concurrent enrollment.) as listed at https://catalog.wsu.edu/General/Academics/Courses/CPT_S.

4 Course Materials

4.1 Required Textbook

M. Weiss, Data Structures and Algorithm Analysis in C++ (4th ed.), Pearson Education, Inc., 2014. ISBN: 9780132847377. (Online access: http://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam/DataStructures.pdf).

4.2 Required Hardware

A laptop adhering to EECS requirements. Please refer to https://school.eecs.wsu.edu/laptop-requirements/.

4.3 Required/Suggested Software

- Operating systems, e.g., Ubuntu Linux 20.04/22.04, macOS, Windows Subsystem for Linux (WSL) [2].
- Virtual Machine software (optional), e.g., VirtualBox.
- Version control system, e.g., Git [3]. We will use it to submit coding assignments.
- IDE (optional), e.g., CLion [4] or Visual Studio Code [5].
- Compiler, e.g., G++ [6].
- Build system, e.g., Make [7].
- Cross-platform build file generator, e.g., Cmake [8]. We will use it to guarantee your coding assignments can be complied/tested across multiple platforms such as macOS and Linux.

[2] https://learn.microsoft.com/en-us/windows/wsl/install
[3] https://en.wikipedia.org/wiki/Git
[4] https://www.jetbrains.com/clion/
[5] https://code.visualstudio.com/
[6] https://en.wikipedia.org/wiki/GNU_Compiler_Collection
[7] https://en.wikipedia.org/wiki/Make_(software)
[8] https://en.wikipedia.org/wiki/CMake

4.4 Lecture Slides/Notes

All lecture slides/notes can be found on Canvas "Home" page or "Modules" page (usually visible after the same class meeting).

5 Grading

5.1 Grading Weights

- Programming Assignments (PA x 4): 32%. They require you to design, implement, and test your own code.
- Micro-Programming Assignments (MA x 5): 25%. Short C++ assignments, where some template code will be provided. You will need to solve a problem by filling out the rest of the template code. This will require that you practice reading someone else's code, and add to it.
- Written Homework (HW x 6): 18%. Concept, algorithm analysis, and math written documents.
- Midterm Exam: 10%, during the 8-th week of this semester.
- Final Exam: 10%, during the final exam week of this semester.
- Participation and Attendance: 5%. You are expected to attend and participate in lectures regularly. I will also randomly take attendance and possibly give in-class quizzes throughout the semester. Attendance will count as 5% towards your overall grade! For absence, please see the university policy for absences. Contact instructor (Section 7) to address your absence under this policy.

5.2 Late Work Grading

Late submissions can still be graded under some circumstances.

5.2.1 Written Assignments (HWs)

For written assignments in the format of Canvas Quiz, there is NO late submission policy, and the submission will be closed after due date: the due date for written assignments is firm.

5.2.2 Coding Assignments (PAs + MAs)

For coding assignments, TA grader will use the following late work policy.

- After the scheduled due date, you will have a 2-calendar-day non-penalty period, allowing any variations of everyone's schedule.
- After the 2-calendar-day non-penalty period, the maximum possible grade is capped by 20% per day (see examples below).

- Maximum 5-calendar-day late submission (i.e., 2-calendar-day non-penalty and 3-calendar-day penalty) is allowed. After that, assignment submission will be closed.
- Example 1: the scheduled due date is Sep 1, 23:59, PT. I submit my assignment at Sep 1, 23:58, PT (1 minute before the scheduled due date). The maximum possible grade I can get is 100 (full points). No action is taken.
- Example 2: the scheduled due date is Sep 1, 23:59, PT. I submit my assignment at Sep 3, 23:58, PT (1 minute before the end of 2-day non-penalty period). The maximum possible grade I can get is 100 (full points). No action is taken. However, the late submission may influence my schedule for following assignments and give me time constraint (I have 3 x PA + 4 x MA + 5 x HW + 2 x exams this semester).
- Example 3: the scheduled due date is Sep 1, 23:59, PT. I submit my assignment at Sep 4, 00:03AM, PT (4 minutes after the end of 2-day non-penalty period). The maximum possible grade I can get is 80 (= 100 - 20, capped by 20). If my submission originally gets 81 points out of 100, it will be capped to 80. If my submission originally gets 79 points out of 100, it will not be changed.
- Example 4: the scheduled due date is Sep 1, 23:59, PT. I submit my assignment at Sep 6, 23:59, PT (last minute of the 5-day late submission). The maximum possible grade I can get is 40 (= 100 – 3*20). If my submission originally gets 81 points out of 100, it will be capped to 40.
- Example 5: the scheduled due date is Sep 1, 23:59, PT. I would like to submit my assignment at 00:01AM, PT, on Sep 7. I found the submission has been closed, and my submission will not be graded.

5.2.3 Exams

There is NO late policy for exams and due dates for exams are firm. To get special accommodation for exams or extension, please contact Access Center (see Section B) or directly talk to the instructor.

5.3 Extension of Assignments

If more time for finishing an assignment is necessary, it is possible to get extension. For example, you have difficulty for finishing assignments on time due to health condition or other emergency condition. In this case, you can request extension with any supporting document such as doctor's appointment records, doctor's note, medical prescription, etc. Request without reasonable supporting document will not considered.

How to request extension: (i) contact TA if your request is to extend less than 7 calendar days (including 7 calendar days), or (ii) contact both TA and instructor if your request is to extend more than 8 calendar days (including 8 calendar days).

See Section 7 for how to get reach out to instructor and TA in a quick way.

5.4 Submission of Coding Assignments

All coding assignments must be submitted by creating a private Github repository and inviting TA joining this repository. On Canvas, in addition, please attach a URL link to your private Github repository for TA grader's access.

5.5 Grading Process

time milestone	grading progress
due date	-
1 week after due	graded, or technical issue preventing grading: check TA's comment
2 week after due	address technical issues in TA office hour

Table 1: Grading timeline for an assignment.

TA is responsible to handle grading of assignments. The entire grading process should be within *around 2 weeks after the original due date*. See Table 1.

If the assignment is still missing after the submission is closed (see Section 5.2 when submission will be closed), then this assignment will be graded as 0 credit.

If there is any technical issue for TA to grade your coding assignment during the 1st week after due date (e.g., TA could not compile your code or re-produce your program successfully), TA will temporarily place 40% credit for your assignment and leave a comment to your submission on Canvas, so it is important to *check your grade* (at least double check your grade after 1 week of the due date) and respond to TA if anything is required for TA to finish grading. In this case, an in-person demo is required, and TA will leave a comment on Canvas to your submission. This in-person demo can be done during a TA office hour (see Section 7.2 for how to sign up on Canvas Calendar).

If your submission has not been graded within 2 week after the original due date, and there is no comment for the next step from TA, please contact TA grader immediately (see how to in Section 7).

During the *2nd week* after the original due date, if you have concern about grading result or believe that a mistake has been made with grading, please leave a comment on Canvas submission, and meanwhile sign up a TA office hour (see Section 7.2) to address your concern. After the end of the 2nd week, grading result will be official.

5.6 Grading Scale to Letter Grade

Table 2 shows the grading scale from percentage to letter grade. Curving grade is possibly considered only if the final grade is very biased and skewed that cannot reflect the performance in this course.

6 Course Schedule

The more detailed course schedule can be found on Canvas "Course Schedule" page or "Home" page, as well as WSU Academic Calendar. The schedule is tentative and subject to change.

94-100.00%	A
90-93.99%	A-
86-89.99%	B+
82-85.99%	B
78-81.99%	B-
74-77.99%	C+
70-73.99%	C
66-69.99%	C-
62-65.99%	D+
58-61.99%	D
0-57.99%	F

Table 2: Grading scale to letter grade

6.1 Class Lecture Schedule (Tentative)

Week 01, 08/19 - 08/23: Course overview and programming tools

Week 02, 08/26 - 08/30: Data structure review

Week 03, 09/02 - 09/06: Math review

- Labor Day–ALL UNIVERSITY HOLIDAY, Monday, September 2, 2024

Week 04, 09/09 - 09/13: Algorithm analysis

Week 05, 09/16 - 09/20: Tree

Week 06, 09/23 - 09/27: Tree

Week 07, 09/30 - 10/04: Tree

Week 08, 10/07 - 10/11: Tree

- Midterm exam, Canvas Quiz, Monday, Oct 7, 2024

Week 09, 10/14 - 10/18: Hashing

Week 10, 10/21 - 10/25: Hashing

Week 11, 10/28 - 11/01: Parallel computing

Week 12, 11/04 - 11/08: Heap

- Veterans Day–ALL UNIVERSITY HOLIDAY Monday, November 11, 2024

Week 13, 11/11 - 11/15: Sorting

Week 14, 11/18 - 11/22: Union and find, graph

Week 15, 11/25 - 11/29: No class, Thanksgiving vacation

Week 16, 12/02 - 12/06: Graph, Final exam review

Week 17, 12/09 - 12/13: Final exam week

7 Contact Instructor And TA

There are many ways to reach instructor and TA. This section includes several efficient ways to do so. Generally,

- regarding assignments and grading result, contacting TA is preferred;
- regarding questions in course materials, contacting the instructor is preferred.

7.1 During or Right After Class

Discussion during class or right after class is always a direct way to communicate with instructor. Technical questions about course materials are welcome.

If you have other concerns/requests (such as assignment extension requests, technical questions) about assignments and grading result, you can also talk with instructor directly right after class and send a follow-up Canvas Message for a confirmation later. In this case, a follow-up Canvas Message is still necessary. See Section 5.3 for requesting extension of assignments, Section 5.5 for addressing your concerns about grading result.

7.2 Office Hour

Instructor and TA will create office hour appointments on Canvas Calendar, so it is convenient to make appointments to organize and participate in discussion during office hour. See Appendix A for how to sign up an appointment on Canvas Calendar. A walk-in office hour is also possible, but not guaranteed, so signing-up a time slot on Canvas Calendar is highly recommended.

7.3 After Class And Office Hour

A *preferred* way to contact instructor and TA after class is to send a Message on Canvas to instructor and/or TA. Check Appendix A about how to send a Message on Canvas to a user in your course. Most Messages on Canvas will be replied within 3 workdays, so that is the recommended/preferred way to guarantee response on time. (Instructor and TA may sometimes miss your message during some busy period of a semester, so please send a second message if not getting response in 3 workdays).

An *alternative* way is to send an email to instructor and TA, but we still recommend using the preferred way for better organization of communication to avoid any missed emails.

8 Academic Integrity Policy

We must ensure that academic honesty is upheld. All graded assignments/exams must be done independently. If you need any assistance, please contact the intructor or TA. Learn more about Plagiarism, Unauthorized Assistance, Fabrication, Acts of dishonesty, etc., by visiting WSU's Academic Integrity Policy.

Particularly, if the submitted assignments are built on any existing work (e.g., a blog, a tutorial, etc.), **a complete and appropriate acknowledgement/citation is required** to identify what the original part your submission has done compared to the existing one. Without a complete and appropriate acknowledgement/citation if built on other sources, the submitted assignments will receive failing grade and will be considered as plagiarism.

8.1 Can I Use Generative AI?

Generative AI, such as ChatGPT, should NOT be used to help write the code for any graded assignments/exams. However, generative AI can be used to search any questions that you have when you attempt complete your assignments, analogous to using Google for searching your general questions, such as "what is Big-O notation?" or "what is key operation in AVL tree?", rather than a specific question like "help me write a BST in C++". Such general questions should not be directly related to the specific assignment questions.

Generative AI must NOT be used during any exams.

8.2 Can I Use Code from Internet?

You may use others' code in parts of your assignments, if you (i) give a complete and appropriate acknowledgement/citation in your submission to the source where you used the code, including the URL links, etc., and (ii) highlight which part of your submission is your original work that is different from the referred source.

However, TA will grade your submissions depending on how much major task you complete by your own. Therefore, using too much external code in your submission can still result in bad grades.

A How to Use Canvas: Tutorials

- Send/reply Canvas Message: this tutorial.
- Canvas appointment sign-up: this tutorial.

B WSU Access Center

Let instructor know if you need special accommodations (such as extended time, make up exams, private or semi-private room for exams, quiet/reduced distraction environment, etc.) for assignments, attendance, midterm exam and final exam as early as possible. You may request special accommodations via visiting WSU's Access Center.