CIS 25 – Programming Using C++
Laney College – Semester: Fall 2023 Code: 41064

Recommended Preparation:	CIS 6 or CIS 61 (Strongly recommended my CIS 6)
Knowledge Should Already Be had	**Variables** **Conditional & Iteration/Loop Structures** **Good knowledge and use of functions** **Good knowledge and use of arrays** **Good knowledge and use of functions & arrays**

Instructor: Tuan T. Nguyen

Contact Info:

Office: Online

Office Hours: Thursday: after class
(All Zoom) Friday: 8:00 pm to 10:00 pm and 1 hour TBA

Email: cis25LaneyEmail@gmail.com

Lecture: ZOOM Live Lecturing
Tuesday & Thursday: 6:30 pm to 7:45 pm

Lab: ZOOM Live Lecturing & Discussion
Tuesday & Thursday: 8:00 pm to 9:15 pm

Students will be notified by the instructor of any changes in course requirements or policies.

Requirements:

- **All students must use cameras and show yourself from your shoulder up—no exceptions!**
- **If there are required face-to-face, in-person, or in-class meetings, then masks and masking will be required for all participants—instructor(s), student(s), and others.**

Be aware of the following behavior and usage:

No Profanity and Foul Language in ALL of my classes – A report to College Administration will be submitted if yo do disturb and use Profanity/Foul Language in meetings (Lecturing, Office Hours, etc.)! Please think and use words/language that are respectable during class meetings.

Students are responsible for checking Important Administrative Dates such as:

1) **Last Day to Drop Without 'W' and Receiving Refund**
2) **Last Day to Change Grading Option (Pass/No-Pass)**
3) **Last Day to Drop Without 'W' AND MUST STILL PAY ALL FEES**
4) **Last Day to Withdraw and Receive a 'W'**
5) **All students are responsible for checking dates of above and paying all fees themselves.**

Lecture Materials:

a. Personal Lecture Notes and Handouts

b. Recommended References:

"C++: Primer Plus" 6th Edition Authors: Stephen Prata Publisher: Addison-Wesley ISBN: 1-4188-3639-7 *"Starting Out With C++ EARLY OBJECTS"* 10th Edition Author: Tony Gaddis, et al. Publisher: Pearson ISBN: 978-0-13-523500-3 *"C++ Programming – From Problem Analysis to Program Design"* 7th Edition Author: D.S. Malik Publisher: Cengage	*"The C++ Programming Language"* 4th Edition Authors: Bjarne Stroustrup Publisher: Addison-Wesley ISBN: 0-321-56384-0 *"C++ Primer"* 3rd Edition Authors: Stanley B. Lippman, Josee Lajoie, Barbara E. Moo Publisher: Addison-Wesley ISBN: 0-201-82470-1

Grade Percentage:

Participation/Attendance 10%

Note!

- You will lose one grade if you missed 4 or more class meetings.
- You will receive a 'D' grade if you missed 8 or more class meetings.
- You will receive an 'F' grade if you missed 10 or more class meetings.

Quizzes/(Tests) 35%
Homework/Programs 30%
Final Exam 25%

Quizzes & Tests:

Tests/Quizzes will be given and posted on Canvas mostly. These quizzes are to be given based on the material presented in class meetings.

There will be **NO MAKE-UP** for any of the tests/quizzes and final exam.

All quizzes and final exam are to be done online and/or under supervised remote settings (under the supervision of the instructor as needed). This will be explained in class.

Programming Assignment Due Dates:

Students must submit their work Students must submit their work by the given date and time (e.g., Sunday on or before 11:00 pm). The work submissions will be through emailing as attachments based on the specified and required formats and structures—for code, only `*.c`, `*h`, and no Zip File and no other file formats!

If the work submission is after 11:00 pm on the given date, then the work submission is late. A late work submission within 24 hours will get a 50 % penalty. Between 24 hours and 48 hours late, the penalty will be 75%. After 48 hours late, the submission will not count regardless of the reasons, and the student will receive zero (0).

Some work may have 5 days to submit, and some may have 10 days or 2+ weeks as specified. All students must attempt and work on the work as soon as it is available on Canvas.

The hints for work are discussed in classes, and code samples are posted on Canvas. Everyone will have the same length of time to submit regardless of reasons. Start to work early, and all work must follow the instructions and formats given in class.

Work format+convention+style will be discussed and provided. Students must follow these formats, conventions, and styles to receive full credit for all work. More information will be available through demonstration in class.

Grading Guideline:

90 – 100% = A 75 – < 90% = B
60 – < 75% = C 50 – < 60% = D Below 50% = F

An "I" grade for incompletion will not be given except in the event of **extremely unusual, unexpected, and unavoidable** circumstances arising at the end of the semester for a student, who is up-to-date with quizzes/examinations taken, homework submitted, and good attendance/participation.

Attendance Policy:

Regular attendance is required. Individual meeting Zoom links will be posted for each meeting → Newly updated Zoom links will be used for individual class meetings.

Attendance plus participation percentage counts toward your grade as above. To join the class Zoom sessions, you must log into Zoom with the correct and required information (complete first name followed by complete last name).

To earn attendance verification, you must show your physical present during the meeting and also provide an attendance email as proof. Missing one of these two steps will result in an absence in the attendance listing for each class Zoom meeting. The "How-To" for attendance verification will be explained in class. Also, office hours do not have attendance lists!

Please do not be late for more than 15 minutes while logging into the Zoom meeting as then you may not be allowed to join the ongoing session. Consequently, you are absent from the class meeting!

Be Warned About Dropping from Class!

I may not drop you from my class at all. However, if you decide to drop my class, you are responsible for doing it. In any case, you should confirm with the registrar's office about your enrollment in a class when in doubt.

As for online meetings,

- All meetings are synchronous online. That means they are live and no recordings!
- For online class meetings, all students must connect using the given Zoom link for each individual meeting.
- Please be on time! If you are 15+ minutes late, you may not be allowed to join the meeting!
- For online meetings, students will sign in with full name that can be verify with class roster before getting admitted to the sessions.
- Capturing of Zoom screens will take place from time to time to verify students.
- Again, all students must use cameras and show yourself from your shoulder up—no exceptions!

Class Etiquette and Rules:

1. Be on time!
2. No cheating is allowed in this class. Cheating will result in a fail grade.

3. No recordings will be done for all class meetings!
4. Submitting work using the required format as discussed in class.
5. You should raise your questions to the whole class unless they are inappropriate! Spend time digesting the materials before asking the questions. All questions have value, but not all questions may receive answers.
6. During Zoom sessions, you may get called to share your understanding of the discussion points. Be ready!
7. Always get help from appropriate sources: Instructors, Classmates, and Others. Please
    - Do not wait until it becomes too late or unsolvable.
    - Use available resources for your learning process and advantage.
    - At all times, you must use the required formats and instructions as given in class discussions.
8. Students are free to collaborate with classmates (and others) in discussing and developing programming logic. However, students must program/do the work by themselves (i.e., **students must write, run and debug exercises/programs themselves**) to fulfill the requirements unless otherwise indicated.
9. No copying (cheating!) of any form is allowed. In my judgment, if any parts of a student's programs were copies from others (or machines, systems), then the copied material will receive no credit. If I cannot determine the origination of the source and its copied material, then both will receive no credit for all material involved. Acknowledge and quote the sources where you got help from; this is highly appreciated.
10. Unless being mentioned as part of a setup, copying is cheating!
11. **No inappropriate discussions during class meetings**.
12. During class meetings, cell phones are for emergency use only! Please be considerate when using the phone during class sessions!

Student Requirements & Observance:

1. **Programming convention and style will be emphasized. You are required to follow the recommended convention given in class.**
2. Writing code is not just the code works. It also involves patience, care, and code idioms+forms along with others. Please see the C/C++ coding convention file(s), the posted code, and the sample work discussed and written in class. Of course, you will see me writing live code in front of you! I assume you should have learned that "Patience is a Virtue!" and “mistakes are the precursors of learning”.
3. To earn attendance+participation credit, you must participate in signing exercise and other activities for every meeting. If you did not attend the meeting, the signing exercise is invalid.
4. Provide an email address that will be used (appropriately) during the semester. Everyone is responsible for checking class messages posted on Canvas or emailed out.
5. **Just before coming to class, please check your Canvas + mail box for updates and messages**.
6. Regarding emailing:

a. Do not send junk mail.

b. Do not spam/phish.

c. Do not send any form of advertisement.

d. Do appreciate for bringing up questions in class, however.

7. When sending an email, you must use the subject line appropriately. A message without proper "Subject Line" will not be read. Always sign your "real name" at the end of your message.

   While sending a message, a proper "Subject Line" must be used—without a proper "Subject Line" your email will not be read and will be deleted. The proper "Subject Line" should have the following format:

   CIS 25 Fall 2023 YourFirstname+Lastname : Reasons/Needs

8. Zoom Lectures & Discussions are used for all meetings/sessions.

Again, for your understanding and agreement

a. Come to class on time; and
b. Do not disturb classmates and class settings; and
c. Do not publish (audio, video, text, etc.) class discussions & stuff (materials and stories) in any platforms. Only faculty should give permission to access to the course; and
d. Always acknowledge the sources of texts and words that you use from; and
e. Be active and participate in class meetings; and
f. Provide thoughtful comments instead of "I agree" or "Simply don't know"; and
g. Enjoy the class!

LEARNING OUTCOMES

1. Outcome: Solve Problems with Computers: Interpret and analyze a business problem and design, code, compile, test and debug a program solution in C++ using proper program syntax balancing efficiency and maintainability using object-oriented programming concepts to create applications solving business problems with classes and objects.

   Assessment: Evaluate lab assignments and project for proper program syntax balancing efficiency and maintainability using object-oriented programming concepts.

2. Outcome: Program Structure: Use data types to define variables and arrays, analyze and construct algorithms and translate to appropriate control structures of sequence, selection and iteration, and design functions to structure programs into smaller, simply defined components

   Assessment: Evaluate quiz and test questions on variables and arrays; algorithms and sequence; selection and iteration; and functions for clear and efficient program structure.

3. Outcome: Development Tools: Use software development tools including editors, libraries and compilers; Use a wide range of troubleshooting methods and tools to isolate and fix bugs and develop User Interface (UI) or program interaction to meet user requirements

   Assessment: Evaluate lab assignments and project for proper use of development tools and User Interface (UI).

Tentative Topics to Be Covered:

1. Computer Programming – A Review

2. Programming Convention & Styles
3. Review – Memory, Arrays, Pointers and Structures
4. C++ Syntax and Structures – Introduction
5. References and Pointers
6. `struct` – Design and Implementation
7. Functions in C++
8. Memory and Functions
9. Function Overloading
10. Dynamic Memory
11. Object-Oriented Programming (OOP) Using C++ – Introduction to Classes
12. C++ OOP Components
13. `class` – Design and Implementation
14. Constructors and Destructor
15. Inheritance/Class Hierarchy – Overview and Introduction
16. Polymorphism – Overloading and Overriding
17. Operator Functions
18. **`friend`** Functions
19. Virtual Functions, Pure Virtual Functions
20. Multiple Inheritance and Virtual (Base) Classes
21. Abstract and Final Classes
22. Template – Functions & Classes
23. Exception and Exception Handling
24. Recursion – Introduction
25. File Input/Output
26. Applications & Miscellaneous Topics

Note that the materials will be presented with emphases on C++ concepts, understanding, and code implementation (in C++)!

Note also that actual topics may change during the course of the semester.

Syllabus Changes

Even with the intention of having a complete course syllabus with information, I reserve the right to update and adjust this syllabus during the semester. I will inform students with the updates.

Keep in Mind:

- **Learning is a process. Making the process fun is the key in learning.**
- **Be persistent and patient with your learning process.**
- **At times if you are quiet, you will hear a lot of things.**

Be aware of the following assessment:

Your individual work will be used to assess your learning and ability to handle the course and to earn the appropriate grade for your work.

Be aware of the following behavior and usage:

No Profanity and Foul Language in ALL of my classes – A report to College Administration will be submitted if you do disturb and use Profanity/Foul Language in meetings (Lecturing, Office Hours, etc.)! Please think and use words/language that are respectable during class meetings.

Requirement (Again):

If there are required any face-to-face, in-person, or in-class meetings, then masks and masking will be required for all—instructor(s), student(s), and others.

++++++++++

The following are excerpts from course information given in recent CS courses at University of California at Berkeley:

Policy on Collaboration and Cheating

Cheating on a homework, lab, or project will earn you the maximum negative grade on that assignment. For example, if you cheat on a project worth 20 points, your grade on that project will be –20. Cheating on an exam, or cheating twice in any way, will earn you an F in the course. I reserve the right to assign an F in the course to anyone who cheats on a project, though I might not exercise it. All incidents of cheating will be reported to the Office of Student Conduct, who will maintain records of your academic misconduct throughout your undergraduate career.

We encourage you to help each other learn the material by discussing the work *before* you do each assignment. Explaining the meaning of a question or offering advice on what a compiler error message means are interactions that we encourage. On the other hand, you should **never** have another student's solution or code in your possession, either electronically or on paper. (We will call this the **No Code Rule.**) If you are not sure whether a particular interaction is appropriate, talk to your TA or the instructor.

If you receive a significant idea from someone in the class, explicitly acknowledge that person in your solution. Not only is this a good scholarly conduct, it also protects you from accusations of theft of your colleagues' ideas.

Presenting another person's work as your own constitutes cheating, whether that person is a friend, an unknown student in this or another class, or an anonymous programmer on the web who happens to have solved the problem you've been asked to solve. Everything you turn in must be your own doing, and it is your responsibility to make it clear to the graders that it really is your own work. The following activities are specifically forbidden in all graded course work:

> Possession (or theft) of another student's solution or partial solution in any form (electronic, handwritten, or printed).
>
> Giving a solution or partial solution to another student, even with the explicit understanding that it will not be copied.
>
> Working together (with someone other than your partner for the assignment) to develop a single solution and then turning in copies (or modified versions) of that solution under multiple names.

…

You will do some of the projects in teams of two or three students. Any assignment that is not designated as a team assignment must be done individually. On team assignments, you share everything with your teammates, but the rules for individuals given above apply to teams. You may not work with another team or share solutions between teams. Each individual in a team is responsible for the entire project, which means that you will be held responsible if your partner uses another team's solution to produce part of your team's solution. Once you've begun coding a project, you may not change the size of your team or exchange partners without our permission. If your team has irreconcilable conflicts after beginning a project, you must speak to me before breaking up or reforming your team. Only one of the new teams (at most) will be allowed to keep the code developed thus far.

Cheating will be policed by advanced cheating-detection software. If you share code with another team, you will be caught, even if you take steps to hide your cheating.

In my experience, nobody begins the semester with the intention of cheating. Students who cheat do so because they fall behind gradually and then panic. Some students get into this situation because they are afraid of an unpleasant conversation with a professor if they admit to not understanding something. I would much rather deal with your misunderstanding early than deal with its consequences later. Even if you are convinced that you are the only person in the class that doesn't understand the material, and that it is entirely your fault for having fallen behind, please overcome your feeling of guilt and ask for help as soon as you need it. Remember that the other students in the class are working under similar constraints—they are taking multiple classes and are often holding down outside employment.

<u>And Another Reminder (below)</u>

"…Before you develop your solutions to each problem you are encouraged to discuss it with other students, in groups as large or small as you like. When you turn in your solution, you must give credit to any other student(s) who contributed to your work. Working on the homework in groups is both a good way to learn and a lot more fun!

…

Working cooperatively in groups is a change from the traditional approach in schools, in which students work either in isolation or in competition. But cooperative learning has become increasingly popular as educational research has demonstrated its effectiveness. One advantage of cooperative learning is that it allows us to give intense assignments, from which you'll learn a great deal, while limiting the workload for each individual student. Another advantage, of course, is that it helps you to understand new ideas when you discuss them with other people. Even if you are the "smartest" person in your group, you'll find that you learn a lot by discussing the course with other students. For example, in the past some of our best students have commented that they didn't really understand the course until they worked as lab assistants and had to explain the ideas to later students."

.....

"Copying all or part of another person's work, or using reference materials not specifically allowed, are forms of cheating and will not be tolerated."

++++++++++

The following is an excerpt from the course information given in a recent CS course at Stanford University:

"… In computer science courses, it is usually appropriate to ask others — The TA, the instructor, or other students — for hints and debugging help or to talk generally about problem-solving strategies

and program structure. In fact, I strongly encourage you to seek such assistance when you need it. The important point, however, is embodied in the following rule:

Rule 1: You must indicate on your submission any assistance you received.

If you make use of such assistance without giving proper credit, you may be guilty of plagiarism.

In addition to providing proper citation—usually as part of the comments at the beginning of the program—it is also important to make sure that the assistance you receive consists of general advice that does not cross the boundary into having someone else write the actual code. It is fine to discuss ideas and strategies, but you should be careful to write your programs on your own. This provision is expressed in the following rule:

Rule 2: You must not share actual program code with other students.

In particular, you should not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it. Similarly, you should not discuss your algorithmic strategies to such an extent that you and your collaborators end up turning in exactly the same code.

Discuss ideas together, but do the coding on your own.

The prohibition against looking at the actual code for a program has an important specific application in computer science courses. Developing a good programming assignment often takes years. When a new assignment is created, it invariably has problems that require a certain amount of polishing. To make sure that the assignments are as good as they can be, Stanford's department—like most others in the country—reuses assignments over the years, incorporating a few changes each time to make them more effective. The following rule applies in all computer science courses:

Rule 3: You must not look at solution sets or program code from other years.

Beyond being a clear violation of academic integrity, making use of old solution sets is a dangerous practice. Most assignments change in a variety of ways from year to year as we seek to make them better.

Each year, however, some student turns in a solution to an assignment from some prior year, even though that assignment has since changed so that the old solution no longer makes sense. Submitting a program that solves last year's assignment perfectly while failing to solve the current one is particularly damaging evidence of an Honor Code violation.

Whenever you seek help on an assignment, your goal should be improving your level of understanding and not simply getting your program to work. Suppose, for example, that someone responds to your request for help by showing you a couple of lines of code that do the job. Don't fall into the trap of thinking about that code as if it were a magical incantation—something you simply include in your program and don't have to understand. By doing so, you will be in no position to solve similar problems on exams.

The need to understand the assistance you receive can be expressed in the following rule:

Rule 4: You must be prepared to explain any program code you submit.

Although you should certainly keep these rules in mind, it is important to recognize that the cases that we bring forward to Judicial Affairs are not those in which a student simply forgets to cite a source of legitimate aid. Most of the students we charge under the Honor Code have committed fairly egregious violations. Students, for example, have rummaged through paper recycling bins or undeleted trash folders to come up with copies of other students' programs, which they then turn in as their own work. In many cases, students take deliberate measures—rewriting comments, changing variable names, and so forth — to disguise the fact that their work is copied from someone else.

> Despite such cosmetic changes, it is easy to determine—and we have tools for doing so—that copying has occurred. Programming style is highly idiosyncratic, and the chance that two submissions would be the same except for variable names and comments is vanishingly small.
>
> …."

++++++++++

The following is an excerpt from the course information given in a CS course at Columbia University (eon ago):

"**Cheating**

Cheating makes me extremely unhappy. It's dishonest, unfair to other students who are doing their own work fairly, and very unpleasant to deal with as an instructor. And since misery loves company, I will do my best to make cheaters extremely unhappy too …"