CIS 27 – Data Structures and Algorithms
Laney College – Semester: Spring 2024 Code: 23224

Please watch "How do computers read code?" — https://www.youtube.com/watch?v=QXjU9qTsYCc

Knowledge Should Already Be had

(Strongly STRONGLY Recommended) **Arrays, Functions & Pointers**

Use of Arrays, Functions & Pointers

Instructor: Tuan T. Nguyen

Contact Info:

Office: Online

Office Hours: Before Thursday Class
(All Zoom) Friday: 8:00 pm to 10:00 pm
1 TBA

Email: cis27LaneyEmail@gmail.com

Lecture: ZOOM Live Lecturing
Tuesday & Thursday: 9:00pm to 10:15pm

Lab: ZOOM Live Lecturing & Discussion
Tuesday & Thursday: 10:30pm to 11:45pm

Final Exam Starting at 9:00 am on Thursday, May 23, 2024

Contact hours and minimum expectations for hours out-of-class

This is a 4 unit class, which meets for 4+ hours perweek. For each hour that class meets, you should expect to do 2 hours of homework outside of class. That means you should plan for 8+ hours of homework each week for this class. Homework includes reading textbook, working on assignments, studying for quizzes and exams.

Zoom Attendance Requirements:

- **Students are required to attend all Zoom meetings.**
- **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 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.

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:

1. Personal Lecture Notes and Handouts.
2. Some books in my possesion → Only for fun reading (if you think so!) and by no mean of endorsement and/or recommendation.

"Data Structures & Program Design Using C – A Self-Teaching Introduction" Author: D. Malhotra and N. Malhotra Publisher: Mercury ISBN: 9781683922070 "Data Structures, Algorithms & Software Principles in C" Author: Thomas Standish Publisher: Addison-Wesley ISBN: 0-201-59118-9 "Algorithms in C – Parts 1-4". 3rd Edition Author: Robert Sedgewick Publisher: Addison Wesley ISBN: 0-201-31452-5	"Data Structures Using C & C++" Author: Langsam, et al. Publisher: Pearson/Prentice Hall ISBN: 9780130369970 "Data Structures Using C++" Author: D.S. Malik Publisher: Thomson, Course Technology ISBN: 0-619-15907-3

Grade Percentage:

```
Quizzes/(Tests)           35%
Homework/Programs         30%
Final Exam                25%
Participation/Attendance  10% (Synchronous/Live Zoom Class Meetings)
```

Note that class Participation/Attendance is very important!

Then, regardless of any percentage that you may have earned from the above breakdown and any circumstances, you will

- *Lose one grade if you missed 5 or more class meetings.*
- *Receive a 'D' grade if you missed 8 or more class meetings.*
- *Receive an 'F' grade if you missed 10 or more class meetings.*

Quizzes/Tests:

Tests/Quizzes will be given and posted on Canvas almost weekly. 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's discretion). This will be explained in class.

Required Tools:

C Compiler
Recommended Compilers: Visual Studion (Windows — I only use Windows), Xcode (Mac)
At Your Own Risks: Any other compilers

Programming Assignment Due Dates:

Students must submit their work Students must submit their work by the given date and time (e.g., Saturday on or before 11:30 pm). The work submissions will be through emailing (to class email address above) as attachments based on the specified and required formats and structures—for code, only `*.c, *.h`, and no Zip File as well as no other file formats accepted!

If the work submission is after 11:30 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:

Attendance is required → Individual meeting Zoom links will be posted for each meeting.

Attendance plus participation percentage counts toward your grade as described 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 as used in the college registration and recorded on the class roster).

To earn attendance verification, you must show your physical present during the meeting. To earn the participation, you must also provide quiz-email. 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 and throughout the semester.

As for online meetings,

- All meetings are synchronous online. That means they are live!
- 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 (Firstname Lastname) that can be verify with class roster before getting admitted to the sessions.
- Capturing of Zoom screens may 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. Submitting work using the required format as discussed in class.
4. 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.
5. During Zoom sessions, you may get called to share your understanding of the discussion points. Be ready!
6. Always get help from appropriate sources: Instructors, Classmates, and Others.

   Please

   - Do not wait until it becomes too late or unresolvable/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.
7. 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.
8. 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.
9. Unless being mentioned as part of a setup, copying is cheating!
10. **No inappropriate discussions during class meetings**.
11. During in-person class meetings, cell phones are for emergency use only! Please be considerate when using the phone during class sessions!

Netiquette Statement

Example

All of those affiliated with the Peralta Community Colleges are expected to practice patience and respect for one another, refrain from jumping to conclusions or judgments, avoid labels, and allow for others to speak in a safe learning environment–both in-person and online (on Canvas).

- When communicating verbally or in writing, think about what you have to say and the potential impact despite your best intentions.
- Show respect for the viewpoints of others who may disagree or see things differently than you.
- Strive to find a balance between speaking, listening, and reflecting.
- Consider all the communication that you produce as a reflection of the way you would like to and/or expect to be perceived.

Comments that cause harm, stress, or embarrassment, and work that offends and infringes on the safe space that this educational class should be, will not be tolerated.

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 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 class meetings! I assume you should have already learned that "Patience is a Virtue!" and “mistakes are the precursors and progress of learning”. Of course, learn quick and avoid the same/similar mistakes!

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. *Student must both attend synchronous class meeting and complete the signing exercise before leaving Zoom properly and correctly to receive participation credit*.

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 information**.
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 27 Spring 2024 YourFirstname+Lastname : Question/Reason/Need`
8. Zoom will be used for all meetings/sessions.

Again, for your understanding and agreement, please

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. Try to provide thoughtful comments instead of "I agree" or "Simply don't know"; and
g. Enjoy the class!

STUDENT LEARNING OUTCOMES

1. Outcome: Data Structures: Design and program data structures such as Linked Lists, Queues, Stacks, Trees, etc.

    Assessment: Evaluate quiz and exam results for correct understanding of data structures such as Linked Lists, Queues, Stacks, Trees, etc.
2. Outcome: Algorithms: Design and program algorithms for sorting and searching, using techniques such as sequential and binary.

    Assessment: Evaluate lab assignments and project for appropriate program algorithms for sorting and searching, using techniques such as sequential and binary.
3. Outcome: Algorithm Efficiency: Evaluate and compare the efficiency of various approaches including best-, worst- and average-case scenarios.

    Assessment: Evaluate quiz and exam results for correct understanding of the efficiency of various approaches to algorithms including best-, worst- and average-case scenarios.

Tentative Topics to Be Covered and Subject to Change:

Item #1

Syllabus:

C Programming Styles and Conventions

Item #2

Introduction to Data Structures & Algorithms – What are Data Structures and Algorithms?
Pointers – Review
Memory – Review
Dynamic Memory – `malloc(), free()`

Item #3

`struct` – Review
Pointers and **`struct`**
Pointers, **`struct`** , and Dynamic Memory
Commenting on Compiler and Coding Issues
Functions – Review

Item #4

Data Structures:
General concepts and applications.
List – Introduction
Array issues and why?
General list structures.
Lists & Linked Lists

Item #5

Linked Lists:
Definition
Operations
Implementation
Applications – Examples

Item #6

Memory Issues:
Dynamic Behavior and Coding in Linked Lists
Recursive Coding, Problems, and Applications
Problems

Item #7

Recursion – Revisited
Why recursion?
Establishing conditions for recursion.
Implementation
Sorting, Algorithms, & Implementation

Item #8

Stack – A data Structure:
Background and why?
Structure specifics and comparison.
Array based discussion
Implementation and applications – examples.
Stacks – Continued:
Linked list based.
Implementation and applications -- examples

Item #9

Queue – A Data Structure:
Background and why?

Structure specifics and comparison.
Array based.
Implementation and applications – examples.

Queues – Continued:
Linked list based.
Implementation and applications -- examples.
Connections between Lists, Stacks, and Queues:
A practice and application discussion
Memory issues

Item #10
Search and Sort – Applications
Trees – Introduction
Trees – Characteristics & Behaviors

Item #11
Binary Tree – Introduction
Definition
Constraints

Item #12
Binary Search Tree (BST) – Introduction
Definition
Constraints
Operations
Implementation
Binary Search Tree (BST) – Structures
Applications – Search, Sort
Traversal
Issues of Binary Search Trees

Item #13
AVL Trees
Constraints
Operations
Implementation

Item #14
Multiway Trees – Introduction
Red-Black Trees

Item #15
Heaps – Structure
Heap Applications – Sorting
Dictionary & Map – Structure
Heap Applications – Sorting

Item #16
Dictionary & Map – Structure
Implementation and Applications

Item #17
Hash Data Structure – Introduction
Hash Variation & Structures
Hash – Collision
Hash — Chaining

Item #18

Graphs – Introduction

Graphs – Revisited

Structures

Applications

Note that the materials will be presented with emphases on concepts, understanding, and code implementation. The required implementation will be in C.

Note also that the actual topics may be changed 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.

Worth to Remember

1. We write code to manipulate data (which ***are*** provided by the user) to produce the required outcome in the most efficient way!
2. Getting the program to work is not enough to earn full credit. Your program must run correctly and follow all proper convention and consistent styles as explained in class in order to receive credit accordingly.
3. Again, writing code is not just the code works. It also involves patience, care and code idioms + forms along with others. Please see the posted code that have been written in class as well as the coding convention C file — posted on Canvas.

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.

Academic Dishonesty Policy

(1) ++++++++++

The following are excerpts from course information given in a CS course 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.

And Another Reminder (below)

"…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."

(2) ++++++++++

The following is an excerpt from the course information given in a 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.

…."

(3) ++++++++++

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 …"

(4) ++++++++++

The information below is found at Peralta

Academic dishonesty policy	Statement on Academic Honesty and Consequences - AP5500 https://web.peralta.edu/trustees/files/2021/05/AP-5500-Standards-of-Student-Conduct-Discipline-Procedures-and-Due-Process-1.pdf Example Any unauthorized assistance on any assignment, including weekly assessments and the final, is considered cheating. Unauthorized assistance includes, but is not limited to, working with other students on these individual assignments, searching for answers on the internet, or using live (or nearly live) assistance through any "tutoring"-type websites. Consequences for Cheating or Other Academic Misconduct: Disciplinary action in the form of progressive sanctions may be taken in cases of violation of the Student Code of Conduct. Such violations include cheating, plagiarism, forgery, alteration of misuse of college documents, records, or identification documents, or furnishing false information to the college. For a list of possible educational sanctions and the Student Code of Conduct violation process, please see: https://web.peralta.edu/trustees/files/2021/05/AP-5500-Standards-of-Student-Conduct-Discipline-Procedures-and-Due-Process-1.pdf