Syllabus-summer 2026

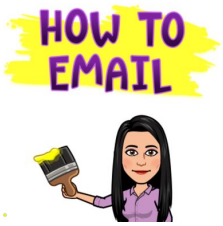
Title of Course-Section:	**CSCI 1230-501 (C Programming)**
Name of Department:	Mathematics, Engineering, & Computer Science (MECS)
Instructor:	Andisheh Dadashi, Assistant Prof. of Computer Science
Email:	andisheh@unm.edu
Credit Hours:	3 credit hours
Class Days/Times:	Fully Online
Class Location:	CANVAS & ZyBook
Office Location:	VAAS-105
Student Drop-in Hours:	Zoom: M 8 a.m. - 9 a.m. (or by appointment)

Note 1: Zoom links related to this course or Student Drop-in Hours can be found on Canvas.
Note 2: The instructor reserves the right to change the syllabus at any point of time during the semester.

Table of content

- Academic Honesty
- Assignments Deadlines
- Chapters of Book
- Course Schedule
- Course Structure
- UNM Deadlines & Academic Calendar
- Disabilities Policy
- Evaluation/Grading Methods
- Important Links
- Late Assignments
- Learning Objectives and Outcomes
- Online Assignments
- QR Codes/Attendance/Absence
- Student Responsibility
- Student Behavior & Collegial Behavior
- How to be Successful in this Course?
- General Support for this Course!
- Teaching Materials
- Technology Requirements
- The Center for Academic Learning
- UNM Valencia Title IX Representative

Contacting your instructor via Email



When emailing me, in the subject of your email, please write down your course name, course number, and section number. For example, the subject of your email to me should be: **CS 1230-501**
You must only contact me with your **UNM e-mail**. Check your **UNM email frequently**. You are responsible for missing any announcement I send via email or posted on Canvas. Failure to identify your message with the class number, and not using your UNM email may result in no response.
Instructor Response Time:
- The best way of contacting me will be via CANVAS Inbox.
- In all cases please, be patient and give me 24 hours to 48 hours to reach back to you.
- I will be available via email (and CANVAS Inbox) during the weekdays until 5 p.m.
- I may not be able to respond to any email on Saturday and Sunday.

Get to know your instructor:

Dr. Andisheh Dadashi is an Assistant Professor of Computer Science, Statistics, and Mathematics in the Mathematics, Engineering, and Computer Science (MECS) Division at the University of New Mexico–Valencia (UNM–VA). She holds a Bachelor's degree in Mathematics and Statistics from Iran, a Master's in Statistical Mathematics from India, and a second Master's in Statistics from the University of New Mexico (UNM). She earned her Ph.D. in Computer Science from UNM Main, where she also serves as a Research Assistant Professor in the Department of Biology, working in Dr. Davorka Gulisija's lab on computational biology.
Dr. Dadashi began her teaching career as a Visiting Lecturer II at UNM–Gallup (UNM–GA), where she collaborated with the Critical Technology Studies Program (CTSP) to recruit and mentor students in research related to critical technology and national security. She has since developed and taught cross-listed data science courses across UNM's main and branch campuses.
Her research focuses on developing theoretical frameworks, computational methods, and statistical tools to study rapid polygenic adaptation—how complex traits evolve in response to changing environments. This work plays a key role in understanding how organisms adapt to climate change.
In recognition of her inclusive teaching, innovative curriculum design, and effective instructional communication, Dr. Dadashi received the 2024–2025 UNM Teaching Excellence Award, jointly sponsored by the Center for Teaching and Learning and the Honors College, as the UNM Branch Campus Teacher of the Year, as well as the 2025–2026 UNM Teaching Excellence Award for Open Communication and Assessment, which honors exemplary instructional communication practices and meaningful assessment strategies that positively impact student learning. These university-wide awards reflect her commitment to accessible, interdisciplinary STEM education.

About CSCI 1230 (C Programming)

CSCI 1230 (C Programming), offers a comprehensive dive into the foundational concepts of data representation, storage, and manipulation using the C programming language. This course is designed to bridge the gap between theoretical data structures and practical implementation techniques, focusing on memory organization and its critical role in computation and efficiency. Students will explore various data storage methods, including linked and contiguous implementations, delve into memory management, and investigate the effects of indices and pointers. Additionally, the course introduces the complexities of the memory hierarchy and its implications on program performance. Through a series of programming assignments in C, students will gain hands-on experience in crafting efficient code and conducting computational experiments to assess the impact of storage design decisions on execution times.

Why C Programming for Data Organization?

C programming language is a cornerstone in the field of computer science, known for its efficiency and control over system resources. Here are several reasons why learning C is an excellent choice:

1. Low-level Manipulation: C provides near-hardware level control, allowing precise manipulation of data storage and memory, which is essential for understanding how data is organized and managed in computing systems.

2. Efficiency and Speed: C is synonymous with performance. Its ability to produce fast executables and efficient manipulation of system resources makes it ideal for demonstrating the impact of different data organization strategies on program performance.

3. Pointer Arithmetic: C's use of pointers is fundamental in understanding how data is accessed and manipulated in memory. This makes it an excellent tool for exploring linked structures, memory management techniques, and the use of indices.

4. Wide Applicability: Knowledge of C programming is crucial in many areas of computer science, from operating systems to embedded systems, enhancing students' career readiness in various tech domains.

5. Control over Memory Hierarchy: C allows students to experience firsthand the challenges and strategies related to managing cache, RAM, and disk storage, which are critical for optimizing program performance.

6. Rich Learning Resources: Despite its steep learning curve, C is supported by an extensive array of learning materials and community forums, which provide valuable support for students navigating through complex concepts.

7. Foundation for Advanced Studies: Mastery in C lays a solid foundation for further studies in more complex and specialized areas such as systems programming, operating systems, and more.

8. Global Relevance: The skills acquired in this course are globally relevant, equipping students with the competencies needed to tackle performance-critical applications and system-level programming across various platforms and industries.

In summary, the C programming language's capabilities in low-level manipulation, efficiency, pointer arithmetic, and control over memory hierarchy make it an exceptional choice for exploring the intricacies of data organization. This course not only enhances technical proficiency but also prepares students for advanced computational challenges in their future careers.

How to be successful in a Programming course?

Success in this course depends on a combination of effort, effective strategies, and a solid understanding of the subject matter. Here are some tips to help you excel in such a course (**What to do**):

1. Attend Classes Regularly: - Make sure to attend all lectures and labs. Consistent attendance helps you understand the material as it's presented, and you won't miss important announcements or assignments.

2. Participate Actively: - Engage in class discussions, ask questions, and seek clarification when you don't understand something. Active participation can help you grasp concepts faster.

3. Read the Course Syllabus: - Understand the course requirements, grading criteria, and important deadlines. This will help you stay organized and plan your study schedule accordingly.

4. Start Early: - Don't procrastinate. Begin assignments and projects as soon as they are assigned. Programming often requires time for trial and error, so starting early allows you to troubleshoot issues.

5. Practice Regularly: - Programming is a skill that improves with practice. Spend extra time coding beyond what's required for your assignments. Experiment with different problems and solutions to reinforce your understanding.

6. Use Online Resources: - The internet is a treasure trove of programming resources. Websites like Stack Overflow, and GitHub can be invaluable for finding solutions to problems and learning from others.

7. Seek Help When Needed: - Don't be afraid to ask for help from your instructor or teaching assistants if you're struggling with a concept or project. They are there to assist you.

8. Understand the Basics: - Ensure you have a solid grasp of the foundational concepts like variables, data types, control structures (if statements, loops), and functions. These will form the building blocks of more advanced programming.

9. Test Your Code: - Always test your code thoroughly before submitting it. Debugging is a crucial skill, and finding and fixing errors is a big part of programming.

10. Study the Documentation: - Get comfortable with reading and understanding documentation for the programming languages and libraries you're using. Documentation is often a programmer's best friend.

11. Stay Inquisitive: - Technology is always evolving. Stay curious and open to new ideas, and tools.

12. Stay Positive and Persistent: - Programming can be challenging, but maintaining a positive attitude and being persistent in your efforts can make a significant difference in your success.

Remember that programming can be challenging, and setbacks are a part of the learning process. Stay persistent, and with dedication and practice, you can excel in this course.

Evaluation/Grading Methods

Your final grade in this class is based on the following components:

Progress Credit	10 %
Participation Activities (PA)	20 %
Challenge Activities (CA)	20 %
Programming Challenges or Labs (PC)	25 %
Final Exam	25 %
Overall	100 %

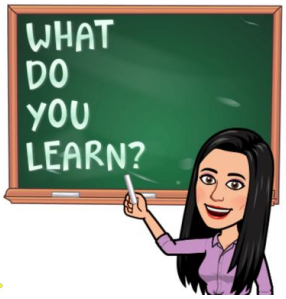
Passing grade is 73% or better. F is a grade lower than 60%
Overall Grades: pluses and minuses may or may not be added to letter grades at the instructor's discretion.
Grades of A+ are not rare and will only be awarded for exceptional work.
In order to pass this course your grade on the final exam must be 70% or better.

This policy ensures academic integrity by allowing the instructor to verify that students fully understand and can explain their work, especially in courses with limited in-person assessments. It helps maintain fairness and ensures the submitted work reflects the student's own knowledge and effort.

Note: The instructor reserves the right to call any student at any time during the semester for an oral discussion or explanation of their submitted work.

Grade	From	To	Grade	From	To	Grade	From	To
A+	98	100	B+	87	89.99	C+	77	79.99
A	93	97.99	B	83	86.99	C	73	76.99
A-	90	92.99	B-	80	82.99	C-	70	72.99

Learning Objectives and Outcomes



Pre-requisites/Co-requisites: 152L with a grade of "B-" or better or 259L with a grade of "C" or better.

Course Description: Data representation, storage and manipulation. Covers the memory organization of data storage and its relation to computation and efficiency. Topics include: linked vs. contiguous implementations, memory management, the use of indices and pointers, and an introduction to issues raised by the memory hierarchy. Programming assignments in C provide practice with programming styles that yield efficient code and computational experiments investigate the effect of storage design choices on the running time of programs.

Goals: The objective of the course is to equip students with a robust understanding of fundamental and advanced programming concepts using C, enabling them to develop efficient and maintainable software solutions.

By the end of the course, the student will be/should be able to:

1. Write and debug C programs that utilize basic constructs such as variables, conditionals, loops, and arrays, applying these tools to solve real-world problems effectively.

2. Design and implement modular programs using user-defined functions, understanding scope, parameter passing, and modular design principles to enhance code reusability and maintainability.

3. Manipulate data structures such as structs and pointers, demonstrating the ability to manage memory dynamically and understand the implications of direct memory access.

4. Apply advanced programming concepts including recursion and pointers to create complex data structures like linked lists and to perform memory management tasks such as allocation and deallocation.

5. Implement and analyze searching and sorting algorithms, using their knowledge of algorithm efficiency and complexity to choose appropriate methods for data handling based on the application needs.

6. Develop proficiency in input/output operations, handling file operations and data parsing to ensure robust user interaction and data processing within software applications.

Technology Requirements

Access to a reliable and fast internet connection is required.

For the course, we use Canvas to navigate through the teaching materials and assignments, but students must also be able to navigate and use other online resources.

Students are required to purchase the online **ZyBook Package** (e-book).

This online book will be on Canvas via **Inclusive Access (IA)**.

ZyBook is the online learning system that accompanies the textbook and includes an e-book. ZyBook is required for the CSCI 1230 course.

How to access Zybook:

- Choose RedShelf on Canvas
- Copy code from RedShelf
- Sign up at zybooks.com
- Enter this code: **UNMCSCI1230DadashiSummer2026**
- Input code from RedShelf
- Click subscribe

A subscription is around $99.

Students are required to download and install GNU Compiler Collection (GCC) on their computer:

How to download GCC:

To learn how to install GCC on a Mac, please watch the following tutorial:
https://youtu.be/HYrXBoDJmcw?feature=shared

To learn how to install GCC on a Windows, please watch the following tutorial:
https://youtu.be/k6juv3mIr9o?feature=shared

You can use any Integrated Development Environment (IDE) for coding, such as Xcode, CLion, etc. For CLion and other JetBrains IDEs, you can download them from the following website. Be sure to use your UNM email to create an account for free access:

https://www.jetbrains.com/clion/download

By Quality Matters: To successfully participate in this course, students must have regular internet access with sufficient bandwidth to stream videos and access course materials on Canvas and ZyBook. High-speed internet is not required; however, a stable connection is essential to ensure smooth access to course content and assignments.

Course Structure - Book

Book and Package:
The online book will be on learn.zybooks.com. Please, read **above** regarding how to subscribe via IA.
A student's grade is determined by points earned out of 100. The following sections give a breakdown of points; any changes in % or assignments will be based on class needs and communicated early.

- Progress Credit (**10%**)

Scan the QR barcode when it is provided to you during class time (lecture videos).
Answer the questions on the QR code form and submit the form before class ends (before the deadline).
Join and participate in tasks and discussions on Canvas.
Work on ZyLabs as a group during class when available, before the due dates.
Credit is given for ongoing progress, incremental achievements, and continuous activity throughout the semester.

- Participation Activities (PA) (**20%**)

– PAs are the section quizzes based on the reading activities from materials in the book
– PAs are due before class
– PAs are not timed
– PAs are posted on ZyBook and must be submitted on ZyBook before the due date

- Challenge Activities (CA) (**20%**)

– CAs are the section quizzes or programming assignments from what you learned during the class
– CAs are due after class
– CAs are not timed
– CAs are posted on ZyBook and must be submitted on ZyBook before the due date

- Programming Challenges (PC) or labs (**25%**)

– PCs are the coding project that is based on the content of each unit
– PCs are due every three to four weeks
– PCs are not timed
– PCs are posted on ZyBook and must be submitted on ZyBook before the due date

- Final Exam (**25%**)

– Exam is programming projects or/and multiple-choice questions based on the covered units
– Exam is timed (90 minutes)
– There are 2 submissions
– Exam will appear on ZyBook at the time we agreed upon.
– Exam will be posted on ZyBook and must be submitted on ZyBook before the due date
- Proctored exam may be require

Note: All the programs submitted on ZyBook will be automatically graded and reviewed for plagiarism.

Assignments Deadlines

Understanding deadlines is crucial to your success in this course. Please read the following information carefully so you don't miss any important due dates.
– **Discussion and Task Deadlines (Canvas):** All discussions and other Canvas-based tasks have their deadlines posted directly in the **Discussion** section of Canvas. Deadlines are enforced by Canvas. As soon as the deadline passes, the discussion or task will **disappear** and you will no longer be able to submit it.
Plan ahead to ensure you complete and submit your work before the deadline.
– **Attendance Deadlines:** Attendance is tracked differently depending on whether you are an **online** or **in-person/HyFlex** student.

For Fully Online Courses

- For courses during the fall or spring, each module includes **two instructional videos** per week. For courses during the summer (8-week), each module includes **four instructional videos** per week.
- Each video contains a **QR code** that appears at the end.
- You must:
  - Watch videos completely.
  - Scan the QR code using your phone.
  - Submit the form that opens after scanning to receive credit for attendance.
- **Attendance is due by 11:59 PM on the Sunday** of the week the module is completed.

For In-Person or HyFlex Courses

- A QR code will be shared during class.
- You must:
  - Scan the QR code **before class ends**.
  - Submit the form linked to the QR code to record your attendance.
- **No late submissions** are accepted. Attendance must be submitted before you leave class or the Zoom meeting.

– **Assignment Deadlines (ZyBook):** Assignments such as Programming Challenges (PC/Labs), Challenge Activities (CA), and Participation Activities (PA) are completed and submitted through **ZyBook**. Here's what you need to know:

- The due dates for these assignments are clearly listed in the **Assignments** section of ZyBook.
- You will typically have assignments due **almost every other day**.
- It is **your responsibility** to check ZyBook regularly and complete your work before the deadlines.

– **Summary:** Please make sure to stay organized, check Canvas and ZyBook regularly, and reach out if you have any questions.

Task Type	Where to Find Deadline	When It's Due
Canvas Discussions / Tasks	Canvas → Discussions Section	As shown in Canvas (disappears after)
Attendance (Online)	Module Videos (QR Code)	Sunday of each module week
Attendance (In-Person/HyFlex)	In Class (QR Code)	Before class ends
ZyBook Assignments	ZyBook → Assignments Section	Almost every other day

Teaching Materials

Q. Where can you find the materials for this class? Canvas and **zyBooks**
Q. Where do you find and submit the assignments? **zyBooks**
Q. Where do you find your grade? **Canvas** (Gradebook) and **zyBooks**
Q. Where do you find the Tasks and Discussions? **Canvas**

Course information including this syllabus, and all the necessary materials and links, etc. will be available via Canvas.

QR codes/Attendance/Absence

• - Please download a free QR scanner on your smartphone. You should scan the QR code that I provide for you during the first 5 minutes or the last 5 minutes of class (Lecture videos).

• - (Not for fully online classes) You are expected to be on time for each class, stay for the entire duration, have the necessary course materials on hand, and participate in the lecture and/or group activities to receive full credit for Progress Credit each day.

• - (Not for fully online classes) Even if you miss a class, you are still expected to complete the assignments, but you will lose points for Progress Credit.

• - (Not for fully online classes) Attendance is mandatory, if you have four consecutive absences (two weeks) without notifying your instructor, or if you accumulate a total of four random absences (including the first day of class), your instructor reserves the right to drop you from the course. (To value the hard work and dedication of other students.)

Late Assignments

Assignment and exam deadlines are firm. Please manage your time wisely to avoid delays. Late assignments will not be accepted **except in the case of a genuine emergency**, at the instructor's discretion.
Please note: To provide some flexibility, the lowest grade from two Programming Assignments (PA or CA), one Lab, and one Discussion will be dropped from the final grade calculation. This policy is intended to accommodate occasional unforeseen circumstances without requiring formal extensions.
– No early exams or make-up exams will be permitted, **except in documented emergencies**. Flight reservations, weddings, vacations, birthdays, and non-NCAA sporting events **do not** qualify as emergencies.
– There will be **no make-up for the final exam**.
Extending assignments for students is not always ideal for several reasons:

1. Maintaining Course Schedule: Extending assignments can disrupt the planned course schedule. Instructors typically design their courses with a specific pace in mind to cover the required material within the available time. Extending assignments may lead to a lag in the curriculum, making it challenging to cover all necessary topics.

2. Developing Time Management Skills: Meeting deadlines is an essential skill for students to develop. By adhering to due dates, students learn how to manage their time effectively and prioritize tasks. Granting extensions too easily can hinder the development of these crucial skills.

3. Fairness and Equity: Granting extensions to some students but not others can create fairness and equity issues. It may lead to resentment among students who met the original deadlines and frustration among those who did not receive extensions.

4. Accountability: Having firm deadlines encourages accountability among students. When assignments have strict due dates, students are more likely to complete them on time, take their work seriously, and strive for quality.

5. Real-World Preparation: Meeting deadlines is a fundamental requirement in most professions. By adhering to assignment due dates in an academic setting, students prepare for the expectations they will encounter in their future careers.

6. Preventing Procrastination: Allowing extensions can encourage procrastination. Students may delay their work with the expectation of getting an extension, leading to last-minute rushes and low-quality work.

7. Maintaining Academic Integrity: Extending deadlines can create opportunities for academic misconduct, such as sharing answers or copying from external sources.

While there are valid reasons to grant extensions in cases of genuine emergencies or extenuating circumstances, it's crucial for instructors to establish clear policies and guidelines for requesting extensions to maintain fairness, consistency, and the educational integrity of the course. In general, extensions should be exceptions rather than the norm to promote responsible time management and a conducive learning environment.

Student Behavior & Collegial Behavior

According to the Code of Conduct as stated in the Policies and Regulations for UNM, student activities that interfere with the rights of others to pursue their education or to conduct their University duties and responsibilities will lead to disciplinary action. This includes any activities that are disruptive to the class and any acts of academic dishonesty. Students are expected to behave in a courteous and respectful manner toward the instructor and their fellow students. Students may be dropped from a class for inappropriate behavior. For more information: Student Code of Conduct

Since we assume you are all adults, we will expect from you, respectful adult behavior. Engaging in disruptive or unruly behavior could result in your being asked to leave, at which time you will be counted absent and a referral will be sent to the Dean of Instruction. Continuing to behave in this way could result in your being dropped from the course. Disruptive or unruly behavior includes but is not limited to:

- texting or talking on your cell phone or Laptop at any time during class,
- continually talking with your neighbor when we are not working on a group activity,
- working on homework from another class,
- refusing to participate in the class activities.
- reading material or watching media not related to this course or at a time that is inappropriate,
- If your learning preferences differ significantly from the course approach, I encourage you to discuss options with me or consider alternate sections that may better fit your needs.
- If you believe there are too many rules and restrictions set for this course and you prefer not to adhere to them, please drop this course.
- If you think this course requires too much work and you do not have sufficient time to dedicate to it, please drop this course.
- If you believe you lack the necessary background knowledge for this course and do not have the time to work on improving it, please drop this course.
- If you are willing to invest enough time to learn from this course, cooperate with your instructor, and ask for help when needed, please stay with us for a great learning experience together.

Your Responsibility

Time required for This Course: Plan to spend a minimum of 9 to 12 hours per week for this class. There is no guarantee you will pass if you dedicate this amount of time, you still need to learn the material and use your time wisely, but those who pass generally are the ones who spend the time needed to do the work to learn the material.

You are responsible for all material covered in this Syllabus and in class, in assigned readings, and on homework assignments. Not all material on tests will necessarily be covered in class but will be in the assignments. The use of cell phones, headphones, etc. is not permitted in class or exams.

EXPECTATIONS: Students are expected to conduct themselves in a polite, courteous, professional, and collegial manner. Cell phones must be set on silent and be out of sight during class. No food or drink is allowed in the computer labs.

Support!

If you have a documented disability, the "Equal Access Services office" will provide me with a letter outlining your accommodations. I will then discuss the accommodations with you to determine the best learning environment. If you feel that you need accommodations, but have not documented your disability, please contact Cheryl Dilger, the coordinator for Equal Access Services at (505)-925-8910 or cdilger@unm.edu.

If you are struggling in this course, do not be afraid to ask for help!
• Student Drop-in Hours: See my Student Drop-in Hours listed at the beginning of this syllabus."Student Drop-in Hours" Feel free to come by or log in for online Student Drop-in Hours (Info on Canvas), or make an appointment to get help.
• Form study groups: You may work together with other members of our class on the Canvas discussion board.
• Tutoring and Student Services: There are various services provided in our Student Services Department. The Math Center at Valencia campus has free tutoring and open labs. CAPS on the main campus also provides tutoring. For more information please check out the following:
Program Support: CAPS, UNM-Valencia Learning Commons (Tutoring): LRC,
TRIO Student Support Services,
Student Learning Support at the Center for Teaching and Learning., and
Pathways to Articulation and Sustainable Opportunities for Students (PASOS).

Academic Honesty

Having academic integrity is paramount to your success in any class. Plagiarism or cheating is not tolerated. Any instance of this will result in a grade of zero for that assignment. Here is the link to the UNM Academic Dishonesty Policy: https://policy.unm.edu/regents-policies/section-4/4-8.html.
The policy states (Subject to Change Without Notice):
Each student is expected to maintain the highest standards of honesty and integrity in academic and professional matters. The University reserves the right to take disciplinary action, up to and including dismissal, against any student who is found guilty of academic dishonesty or who otherwise fails to meet the expected standards. Any student judged to have engaged in academic dishonesty in coursework may receive a reduced or failing grade for the work in question and/or for the course.

Academic Dishonesty is defined as:
https://pathfinder.unm.edu/campus-policies/academic-dishonesty.html
Academic dishonesty includes, but is not limited to, dishonesty in quizzes, tests, or assignments; claiming credit for work not done or done by others; hindering the academic work of other students; misrepresenting academic or professional qualifications within or without the University; and nondisclosure or misrepresentation in filling out applications or other University records.
Cheating students will be disciplined according to University guidelines. Students should get acquainted with their rights and responsibilities as explained in the Student Code of Conduct https://grad.unm.edu/aire/academic-integrity.html

UNM Valencia Title IX Representative

Title IX (9) Statement:
In an effort to meet obligations under Title IX, UNM faculty, Teaching Assistants, and Graduate Assistants are considered "responsible employees" by the Department of Education
(see pg. 15 - http://www2.ed.gov/about/offices/list/ocr/docs/qa-201404-title-ix.pdf).
https://ceeo.unm.edu/programs/title-ix/index.html
This designation requires that any report of gender discrimination which includes sexual harassment, sexual misconduct and sexual violence made to a faculty member, TA, or GA must be reported to the Title IX Coordinator at the Office of Equal Opportunity (oeo.unm.edu). For more information on the campus policy regarding sexual misconduct, see: https://policy.unm.edu/university-policies/2000/2740.html

Important Links

Disabilities Policy: Office of Equal Access:
Contact the Office of Equal Access at (505)-925-8560 to schedule an appointment.
https://valencia.unm.edu/students/advisement/equal-access-faqs.html
The Center for Academic Learning:
The Learning Center is open Monday – Friday with evening hours Monday – Thursday To schedule an appointment or for additional information call (505)-925-8907
https://valencia.unm.edu/campus-resources/learning-commons/index.html
UNM Valencia Registrar's Office:

Contact Registration Office by calling (505)-925-8580 http://valencia.unm.edu

UNM Deadlines & Academic Calendar:

UNM Deadlines: https://registrar.unm.edu/semester-deadline-dates/index.html .....*And*....
Academic Calendar: https://hr.unm.edu/calendars

Topics

Topics	In detail
1. Introduction to C:	- Basics of Programming - Programming Environment and History - Problem Solving and Debugging Techniques
2. Variables, Assignments:	- Variable Types and Operations - Control Structures and Error Handling - Data Type Conversions and Functions
3. Branches:	- Basics of Conditional Logic - Advanced Branching Techniques - Error Handling and Debugging with Branches
4. Loops:	- Loop Fundamentals - Advanced Looping Techniques - Loop Controls and Variable Scope
5. Arrays:	- Array Basics and Operations - Advanced Array Techniques - String and Character Manipulations
6. User-Defined Functions:	- Function Basics and Benefits - Advanced Function Design - Error Handling and Scope Management
7. Structs:	- Basics of Structs - Using Structs with Arrays and Functions - Project Organization with Structs
8. Pointers:	- Pointer Fundamentals - Dynamic Memory Management - Advanced Pointer Usage
9. Input / Output:	- Standard I/O Operations - Advanced I/O Formatting and Parsing - File Handling

Each of these topics is crucial for building a strong foundation in C programming. Depending on your level of expertise and the specific applications you're interested in, you can delve deeper into any of these areas to gain a more comprehensive understanding of C language.

Schedule Summer 2026

First day of semester: June 1st & Last day of semester: July 28th
Holidays: Junteenth June 19th/ & Independence Day Observed July 3rd

Week-Day	Schedule subject to change if necessary
W1-Module1	Unit 01. Introduction
W1-Module2	Unit 01. Introduction
W2-Module3	Unit 02. Variables and Assignments
W2-Module4	Unit 02. Variables and Assignments
W3-Module5	Unit 03. Branches
W3-Module6	Unit 04. Loops
W4-Module7	Unit 04. Loops
W4-Module8	Unit 05. Arrays
W5-Module9	Unit 06. User-Defined Methods
W5-Module10	Unit 06. User-Defined Methods
W6-Module11	Unit 07. Structs and functions
W6-Module12	Unit 08. Pointer basics
W7-Module13	Unit 08. Pointer basics
W7-Module14	Unit 09. Input / Output
W8-Module15	Unit 09. Input / Output

Due Date Final Exam: 23rd July, Thursday

Note: Be aware that there are assignments due every few days.
Note: Please find the due dates for PAs, CAs, and in-class labs on ZyBook.