

# Comp 524: Programming Language Concepts

## General Course Info

Term: Fall 2024  
Department: COMP  
Course Number: 524  
Section Number: 001  
Time: TR 12:30-1:45pm  
In-Person Location: FB 009  
Piazza: <https://piazza.com/unc/fall2024/comp524>

Zoom Lectures:  
<https://unc.zoom.us/j/98434671245?pwd=eDNJUHZINUxiS1lpTUhOanhTYUVwZz09>

Zoom Professor Office Hours: <https://unc.zoom.us/j/97902404607>

Zoom TA Office Hours: <https://unc.zoom.us/j/98762626789>

Google Drive Folder:  
[https://drive.google.com/open?id=1dT89dIxxB0rz\\_gu8lFkNB7ft1T3ddygP&usp=drive\\_fs](https://drive.google.com/open?id=1dT89dIxxB0rz_gu8lFkNB7ft1T3ddygP&usp=drive_fs)

## Bulletin Description

### **COMP 524. Programming Language Concepts. 3 Credits.**

Concepts of high-level programming and their realization in specific languages. Data types, scope, control structures, procedural abstraction, classes, concurrency. Run-time implementation.

### **Rules & Requirements**

**Requisites:** Prerequisite, [COMP 301](#) or [COMP 401](#); a grade of C or better is required in all prerequisite courses.

**Grading Status:** Letter grade.

.

## Course Description

The objective of this course is to introduce students to the fundamental concepts that differentiate alternative programming paradigms. These alternatives emerged initially in different languages, but some of them have been combined in the same language. The specific languages/systems/libraries that will be used to illustrate new programming paradigms are Lisp and Java lambda expressions (functional programming), Prolog (logic programming), ML (type inference), and Weka clustering and decision tree (programming by example/machine learning)

The course is programming-intensive, and there will be a new assignment about every two weeks. The runtime behavior and source code of each assignment will be thoroughly graded.

Class participation will be another important aspect of this course, and as the class structure describes, the goal will be broad participation from the class rather than participation from a small number of “usual suspects. “

Yet another distinguishing feature of the class will be the large amount of extra credit given, which should allow you to recover from bad exam performance.

Some in-class, graded exercises may be done to help students familiarize themselves with new programming editors and languages.

## Target Audience

The target audience is students wishing to gain a broad understanding of diverse fundamental programming paradigms, with a focus on their design and implementation rather than the nuances of how to use the latest combination of these fundamental concepts to build complex programs.

## Prerequisites

The pre-requisites are basic knowledge of Java, recursion, data structures and design patterns. UNC Comp 401/301 and 410/210 cover these topics.

## Goals and Key Learning Objectives

The course has several related goals: (1) Introduce new languages/paradigms that are alternatives to those seen so far in Java including functional programming, logic programming, type inference concurrent programming, programming by example,

and machine learning. (2) Provide an implementation-level understanding of the differences between the various paradigms. (3) Provide an implementation-level understanding of some of the semantics of functional languages.

A non-goal of the course is to teach specific programming languages, which is why the course is called **programming language concepts** rather than **programming languages**. As an example, we will not learn the Go programming language, which is based on a language called CSP (Communicating Sequential Processes) invented in the 70s. In fact, the inventors of Go are from that era. As the name suggests, the language's distinguishing feature is message passing – a subject of comp 533 (distributed systems) – and it does not offer a new programming paradigm. While you will be learning specific languages to concretely illustrate the paradigms, you will not learn two languages that offer similar paradigms.

## Planned Assignments

The assignments are layered around four themes:

1. social distancing applications, implemented using Java, Prolog, SML, and Weka Decision Trees.
2. implementation of an interpreter of a functional language (called Lisp++) based on Lisp ,
3. use of Lisp/Lisp++ to display recursive lists.
4. use of concurrency abstractions to implement concurrent HPC (high performance computing) applications.

Because of the amount of extra credit features we are willing to grade, it is possible to add enough extra credit to gain a high (more than 110%!) grade or overcome an unexpected exam performance.

## Programming Environment

We will use the Eclipse programming environment even though many of you are unfamiliar with it. The reason is that unlike other environments such as IntelliJ and Visual (Code) Studio it is an open-source environment – perhaps one of the most striking open-source project. This means plugins exist for this environment that do not exist for the others. We will be using plugins for Lisp, Prolog, SML and Java style checks. Some of these – in particular the Prolog plugin – was developed for an older Eclipse version, which also supports the other plugins. So it is recommended you use this version. A Piazza post will direct you to it. **You are free to use some other programming environment or even the command-line and a text editor, but**

**are likely to be less productive if you do so.** We will answer programming environment questions only about Eclipse.

## Instructor

Name: Prasun Dewan  
Office: Brooks 150  
Email: [dewan@cs.unc.edu](mailto:dewan@cs.unc.edu)  
(Use email only if no response from Piazza private post)

Phone: 919 590 6123  
(Very unlikely phone will be picked)  
Office Hours: See Pinned Piazza Post

## Teaching Assistant

Name: Mason Mcphail Laney  
Email: [mlaney@cs.unc.edu](mailto:mlaney@cs.unc.edu)  
(Use email only if no response from Piazza private post)  
Office Hours: See pinned Piazza Post

## What's in a Name:

*I wince if someone addresses me simply as "Dewan", which some students have done in recent semesters*

The following ways of addressing me are fine and I am used to them:

Prasun

Mr/Dr/Prof. Dewan

And in case you are worried about mispronouncing my first name, it rhymes with Bassoon.

## Lecture Mode

**Student-Hybrid Mode:** During the lockdown, I experimented with a Zoom-based student-hybrid mode of learning, which, in comparison to my traditional in-class teaching, increased interactivity manyfold.

Because of the increased interactivity, I have retained this structure even when the lockdown ended and in-person classes started. I call this the **student-hybrid** mode as students *but not the instructors* have the flexibility of not attending physically. However, I may have to leave town at times, and will try to give remote lectures when possible.

This mode will be supported in this course. **This means you must bring a laptop to class and log in to the Zoom session with your speaker and mic muted.**

We will probably have some in-class exercises at preannounced dates. You are free to do them remotely, but if you need help, you should come to class. The exercises will have to do with installing and using unfamiliar software, so you are strongly advised to come in person to class.

All lecture zoom sessions will be recorded mainly to get textual transcripts of the interaction so we can reflect on and improve the lectures. These transcripts will be anonymized before we look at them. Audio and video data will not be downloaded or shared and will disappear from Zoom after 30 days.

## Office Hour Mode

Zoom will also be used for all office hours visits. You will need to let us know at least one hour before an office hour if you plan to visit. A Piazza message explains exactly how. You should visit the TA for assignment problems and the professor for conceptual problems. The TA may not be conversant with the concepts taught in class and the professor may not be familiar with subtle hard to catch mistakes in installing and using software and implementing the assignments.

All office hour zoom sessions will be recorded by default mainly to get textual transcripts of the interaction so we can reflect on and improve the visits. These transcripts will be anonymized before we look at them. Audio and video data will not be downloaded or shared and will disappear from Zoom after 30 days.

**If you feel even slightly uncomfortable with the idea of a Zoom recording of an office hour visit, please let us know, and we will understand and disable recording for you.**

## Textbooks and Resources

The textbook is available online: [Programming Language Pragmatics](#). This was the prescribed book before I taught this course, and it is the most relevant book given what I cover.

PPT Recorded Slides of my lectures are in the lectures subfolder of the google drive folder. Just go into the presentation mode and it will play the recorded lecture from the current slide number. Escape out of the presentation mode to stop the playback.

Some of the other relevant resources are:

[Installing CLisp](#)

[Eclipse Dandeloin Plugin](#)

[LISP Tutorials](#)

[The Lisp Interpreter](#)

[\(How to Write a \(Lisp\) Interpreter \(in Python\)\) - Peter Norvig](#)

[GitHub - rui314/minilisp: A readable lisp in less than 1k lines of C](#)

[Learn C • Build Your Own Lisp](#)

[Lisp interpreter in 90 lines of C++ • GitHub](#)

[Installing C](#)

[Installing Eclipse C++ Plugin: CDT 9.4 \(Eclipse Marketplace\)](#)

[Installing SWI-Prolog](#)

[Eclipse Prolog Plugin](#)

[Learn Prolog Now!](#)

[A Gentle Introduction to ML,](#)

[Programming in Standard ML,](#)

[Installing SML-Site-1](#)

[Installing SML-Site 2](#)

[Installing Eclipse ML Plug-In](#)

[Weka Download](#)

Instructor-provided notes, PPT slides, and videos on the material covered in class will be accessible from the shared Google Drive/YouTube. They will not be posted on Sakai, which will be used however for submitting programs and any quizzes that are given.

## Key Dates

**Midterm:** Tue Oct 15th, in class (two days before Fall Break)

**Final:** Monday, Dec 9, 4pm-7pm

### Attendance Policies

Students are required to attend each class unless there are extenuating circumstances. If such circumstances occur, you should access the class material posted for missed classes, and contact classmates to become aware of the announcements that were made.

You will not get Q/A credit for classes you do not attend.

You need to join the lectures Zoom link to interact in class, even if you are physically attending.

Whether you physically attend is up to you. However, we strongly recommend physical attendance.

## Grading

A grade will be assigned based on performance on homework programming assignments, class participation, and exams. Here is the breakdown:

### Regular Credit

Midterm	25%
Final	30%
Assignment Regular Credit	40%
Class Participation (My QA diary)	5%

### Extra Credit

Assignment Extra Credit	Assignment-specific
-------------------------	---------------------

The weightage given to each assignment will be proportional to the number of points allocated to it. The initial assignments will involve less work and carry less weight also.

### Fudge Factor

I reserve the right to apply a 5% fudge factor for borderline grades to give consideration to other factors such as extraordinary grade improvement and quality of in-class class participation. Please do not demand fudge factor grades - they will be left to my discretion.

### Class Participation

Class participation has to do with questions posed **by the instructor** that **you** tried to answer to the whole class.

Often I echo the questions asked by the students, and these questions do get added as instructor questions.

**You do not get class participation credit for questions you asked and I or someone else answered**

You will maintain a diary listing responses to my questions. A Piazza message explains exactly how.

Responses do not have to be correct but should show some attempt to think about the answer. Every response will get the same credit, there will be no judgment regarding its correctness. A response saying you have no idea on how to answer the question is not a valid response and should not be put in the diary. You should also not put in the diary a response made after the zoom discussion on the question closes. Zoom chat records will be made available to help you compose the diary entry

Each diary will be graded based on the number of entries in it. An automatic tool will extract this number. Instructors may check the validity of the contents. It is of course an honor court violation to deliberately add spurious entries. Spurious entries include fabricating answers not articulated by you to the whole class in My Q/A.

The max score will be based on the number of questions asked. Thus, if you add 15 entries for class QA diary, and 3 entries of My QA (My Questions and Answers) diary, and the number of questions asked is 30, then you get 50%. You should fill the diary for a class by the end of the day the class was held, when things are still fresh in your head. Otherwise, you may or may not get credit for it - depending on when we grade the diaries. At this point we do not know the frequency of grading.

Usually, the answer is given on the slide that follows the one that poses the question. You are free to read the slides before class. But during class, you are expected to compose the answer on your own instead of looking it up from the next slide. As in the past there have been many instances of such lookup, I am thinking of not releasing the PPT deck before class.

Not all questions will be asked using Zoom, we will also have interactive questions for which you get no direct credit.

## **Assignment Grading**

Both source code and runtime behavior will be graded using automatic tests, whose mistakes will be corrected by manual grading. Late penalty is computed automatically by our tests.

Automatic tests of final work will be done using the Gradescope server. We will also allow the server to be used to grade in-progress work – there will be no limit on how many times you can submit to Gradescope.



Automatic tests can also be optionally executed locally on your computer using research tools we have developed. which log your interaction with them anonymously so we can improve and evaluate them. **If you feel uncomfortable with this, please do not run them locally – we allow you to submit to Gradescope an arbitrary number of times.**

Assignments are of different weights determined by their max scores. For instance, an assignment worth 10 points (regular credit) has a weight 1/10th of an assignment worth 100 points (regular credit). Your total assignment grade is the sum of all of your individual assignment scores divided by the sum of all assignment max scores. By doing extra credit, you can get more than 100% on the assignments.

## **Late Penalty**

Assignments are due at 11:55 pm on each specified due date. Homework assignments will be penalized 10% up to one week late and 30% for one to two weeks late. They will not be accepted more than two weeks later. **No assignment can be submitted after the last day of class.**

## **Early Submission Extra Credit**

If you submit by the early submission date of the assignment, you will get 5% extra credit, that is, your score will be multiplied by 1.05. Aiming for early submission extra credit will ensure you do not incur late penalty. On the other hand, submitting later does allow for more extra credit, and many students have chosen to do so.

## **Workload**

The time you spend on the course depends on many factors such as how difficult to track your specific programming mistakes are. The average student should spend 9 hours outside lectures on regular credit portions of the homework assignments.

Many courses have a smaller workload. The amount of code you will write will be small but dense. The assignments will come at regular intervals so you will need to find time throughout the semester to work on assignments.

Not all students are able to do that, especially in their senior year. In general, as in some other classes, the number of students who submit assignments dwindle as the semester progresses, with some dropping the course. This happens mainly due to time management issues – there is enough instructor support to help you finish an assignment.

The early submission early credit should encourage good time management. The late penalty is not as harsh as in some other courses, please use it only when necessary!

## Exam Nature

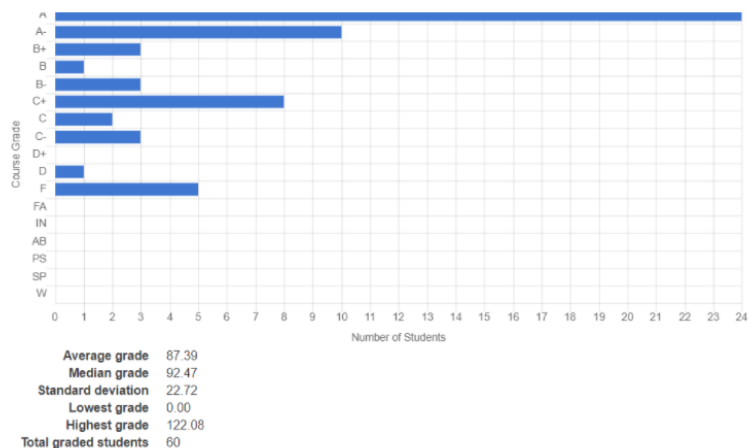
It is unlikely you will be asked multiple- choice questions. Instead, exam questions will likely mirror the questions and discussion in class. Past exams are available from the exams subfolder of the class google drive folder.

## Mapping between Grades and Percentages

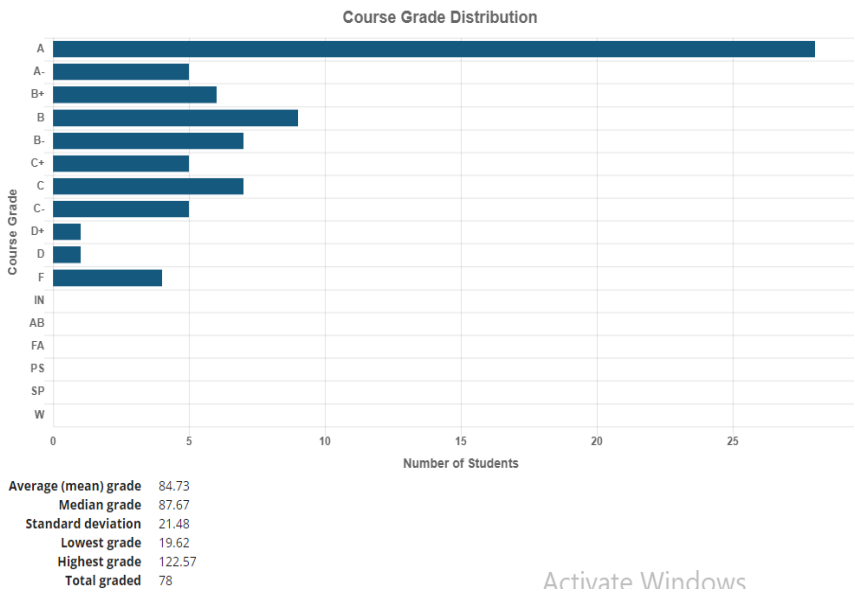
*There is no fixed mapping between the overall percentage and final grade.* Here is the distribution of grades in Fall 2020, 2021, 2022, and 2023.. Just as past performance in stocks is not a predictor of their future performance these numbers are not indicative of how grades will be assigned this year because of differences in the nature of exams (open book vs proctored closed book) and their contents and the extent to which students drop. **The lesson is if you allocate time well to the course, you will get a good grade, and if you do not, you may get a bad grade.** This is how it should be!

### Fall 20

Grade	Minimum %	Remove
A	95.4	Remove
A-	91.01	Remove
B+	86.62	Remove
B	83.79	Remove
B-	81.49	Remove
C+	75.59	Remove
C	72.82	Remove
C-	66.7	Remove
D+	61.2	Remove
D	59.71	Remove



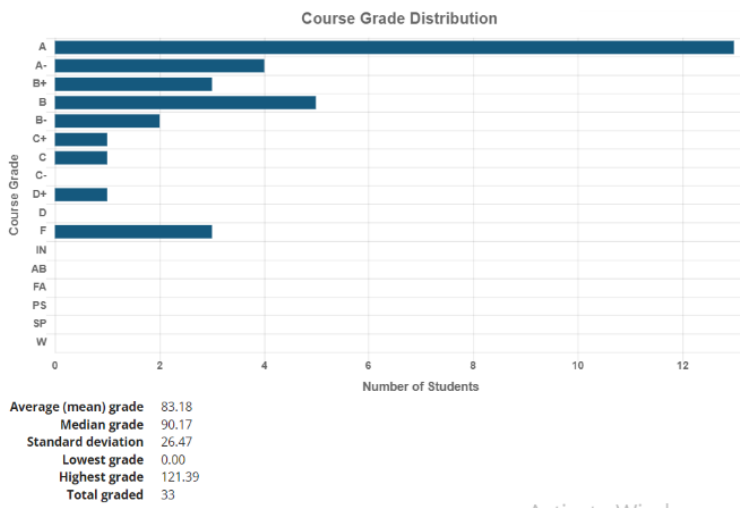
Fall 21



Activate Windows

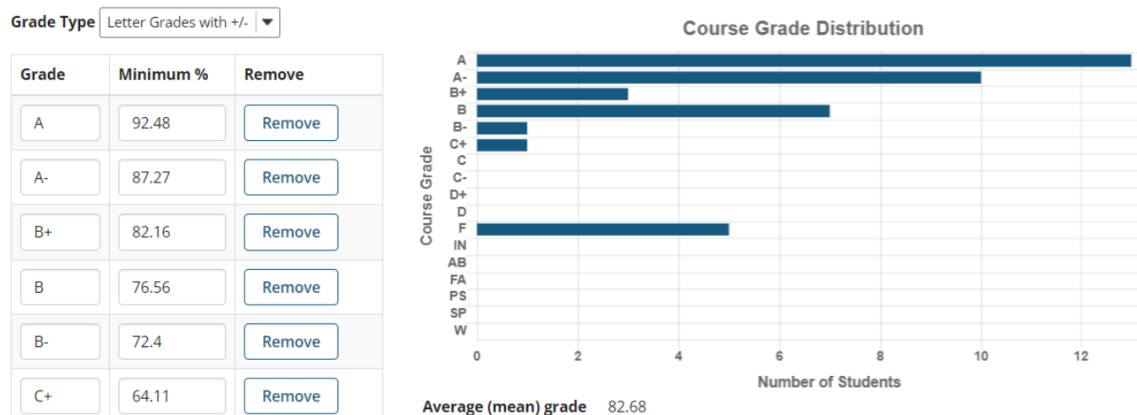
Grade type Letter Grades with +/-

Grade	Minimum %	Remove
A	96	Remove
A-	90	Remove
B+	86	Remove
B	77	Remove
B-	71	Remove
C+	65	Remove
C	58	Remove
C-	55	Remove
D+	52	Remove
D	50	Remove



Activate Windows

## Fall 2023



## Course Policies

Students are required to attend each class unless there are extenuating circumstances. If such circumstances occur, you should access the class material posted for missed classes, and contact classmates to become aware of the announcements that were made.

## Use of AI/LLM

The jury is out on whether the use of AI in coding/debugging assignments promotes or hinders learning. See for example: [From "Ban It Till We Understand It" to "Resistance is Futile"](#): We do not pretend to know the answer. We will leave it you on how you want to use it to complete assignments. However, we require you to document each use of an AI engine for coding a 524 assignment a Piazza post. This post should document the AI system used (e.g. Gemini/GPT/CoPilot). It should have the tag(s) of the assignment(s) for which help was sought, and also the special tag: AI\_Used. Not acknowledging this use will be considered a violation of the honor code, as mentioned below. If you managed to suppress code in interactions with the AI tool, you are free to make the post public; otherwise make it private.

Based on the posts we see on an assignment, we might ban certain kinds of used of AI for future assignments and/or provide special tools to use AI in certain ways.

## Honor Code

You are encouraged to discuss the assignments with fellow students and help them debug programs but required to write/code the solutions/programs individually. **Also, you cannot use solutions from previous offerings of the course.** You cannot make incorrect diary reports or plagiarize the diary text. You cannot simply copy quiz answers from another student in this or a previous offering though you are encouraged to discuss them with others. Making your assignment code public,

through say GitHub, is essentially sharing your code and thus not allowed. You cannot use AI to solve programming problems without documenting such use in a Piazza post, as outlined above.

Not following these rules is a violation of the honor code policy

## Communication with Instructors

See Piazza post on this issue.

## Tentative Course Schedule

Here is an ordered list of topics covered by the course.

1. Why programming languages
2. Hello world in different languages.
3. Comparison of C and Java.
4. Introduction to MVC, Factories in Java.
5. Comparison of and introduction to the following languages using a grader example:
  - a. Decision Tree/Weka.
  - b. Prolog
  - c. SML
  - d. Lisp
6. Semantics of and implementation of the following features Lisp++ to be implemented in the assignments:
  - a. Recursive S-Expressions.
  - b. The Lisp basic functions ([quote](#), [list](#), [eval](#), [load](#), [cond](#), [<](#), [>](#), [<=](#), [>=](#), and, or, not )
  - c. A registrar of operations that add to the basic Lisp functionality.
  - d. The Lisp setq operation with only global scopes.
  - e. Evaluation of atoms and lambda expressions.
  - f. Application of lambda expressions.
  - g. Function expressions and their application.
  - h. Lisp funcall.
  - i. Lisp higher-order functions.
  - j. SML-inspired let expressions
  - k. SML-inspired curry operator
  - l. Lisp defun and SML-inspired defCurry
  - m. funcall of quoted symbols
  - n. setq in non-global scopes
  - o. free variables in functions
  - p. Eager evaluation of list components and Java concurrency constructs need to do such evaluation.
  - q. Eager evaluation of and/or operators (extra credit)

- r. Sockets in Java and Bash needed to implement a distributed and collaborative interpreter.
  - s. **Eager evaluation with thread pool.**
  - t. **Lazy evaluation of lists and functions.**
7. Parameter passing in functions/procedures.
  8. Creation and synchronization of threads.

## Disclaimer

The professor reserves the right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible.