

# CS 3360: Programming Language Concepts

## Fall 2024

CRN: 15001

Lecture: TR 3:00 PM - 4:20 PM in CCSB G.0208

Instructor: Yoonsik Cheon (ycheon@utep.edu); office hours: TR 1:30 PM - 2:30 PM in CCSB 3.0606

Teaching assistant: TBA

Prerequisite: CS 2302 with a grade of C or better (Recommended: CS 3331 and CS 3432)

### Course Objectives

In this course, we will explore the world of programming languages, focusing on both theoretical foundations and practical applications. Our primary goal is to equip you with the tools needed to critically evaluate and quickly master various programming languages and constructs.

- We will maintain a balance between theoretical understanding and hands-on practice. You will survey various constructs and capabilities found in modern programming languages, examining design trade-offs and implementation considerations.
- By understanding the broad spectrum of language features and design alternatives, you will be well-prepared to adapt to new programming languages throughout your professional career. This knowledge is essential for navigating the challenges of learning and using new languages.
- The course will lay a solid foundation for advanced studies in areas like compilers and programming language semantics, providing a deeper insight into the implications of design choices.
- To illustrate general programming language characteristics, we will explore several languages in detail:
  - Dart: A modern object-oriented language.
  - Haskell: A functional programming language.
  - PHP: A web scripting language.
  - Prolog: A logic programming language.
- You will undertake a programming project in each of these languages, giving you practical experience with different programming paradigms and reinforcing your learning through hands-on practice.

By the end of this course, you will have a deep understanding of various programming languages, the ability to evaluate and learn new languages efficiently, and practical experience with multiple programming paradigms.

### Textbooks

Our required textbook for this course is *Concepts of Programming Languages* by Robert W. Sebesta, 12<sup>th</sup> edition, published by Addison Wesley in 2018. This book is available at the UTEP bookstore and is crucial for the course, as reading assignments will be assigned from this book. Additionally, we suggest the following supplementary books:

- Kevin Tatroe and Peter MacIntyre. *Programming PHP: Creating Dynamic Web Pages*. Fourth edition, O'Reilly, 2020. Focus on Chapters 1-6.
- Gilad Bracha. *The Dart Programming Language*. Addison-Wesley Professional, 2016.
- Will Kurt. *Get Programming with Haskell*. Manning Publications, 2018. Specifically, Units 1, 2, and 4.

Electronic versions of these recommended books are available for free through the UTEP Library. If you are accessing the library from outside the UTEP domain, you may need to use a VPN. These textbooks and resources will support your learning by offering a blend of theoretical knowledge and practical applications, aligned with the course objectives.

### Examinations

This course includes both a mid-term exam and a final exam. The mid-term exam is scheduled during a regular class session and will last for 80 minutes, while the final exam aligns with the university's specified date. A makeup exam will be considered only in exceptional or unavoidable situations, such as incapacitating illness or attendance at a

conference for presentation purposes. Should you encounter such circumstances requiring a makeup exam, it is imperative to promptly inform the course staff. For those planning to attend a conference for a paper presentation, arrangements for the exam must be made well in advance. Regardless of the situation, eligibility for a makeup exam is contingent upon providing an official document that explains your circumstances.

### Assignments

This course comprises three distinct assignment types: reading, homework, and programming. It is advisable to allocate approximately 3-4 hours per week for reading and homework assignments, along with an average of 3 hours per week dedicated to programming tasks. Each programming assignment is anticipated to require 8-10 hours of effort, potentially resulting in a more demanding workload during weeks when programming assignments are due.

- Reading assignments: These assignments involve delving into the textbook to prepare for upcoming lessons. Following these readings, quizzes will be administered to ensure the completion of weekly reading tasks and comprehension of core concepts from recent lessons. Typically, quizzes will take around 10 minutes each and will cover material assigned for upcoming lessons as well as selected content from prior sessions. It is essential to note that there will be no opportunities for makeup quizzes, as answers will be accessible after the designated due dates.
- Homework assignments: These assignments entail completing exercises from the textbook's chapters. These assignments are designed to reinforce the material covered in lectures and provide additional practice. Assignments involving material not covered in lectures will be generously evaluated to encourage independent learning and exploration of the subject matter.
- Programming assignments: These assignments provide hands-on experience in specific languages and programming paradigms. Throughout the course, you will work with PHP for web scripting, Dart for contemporary object-oriented programming, and Haskell for functional programming. These assignments are crucial for developing practical skills and understanding the real-world application of programming concepts.

It is mandatory to complete all assignments individually unless explicitly stated otherwise. While engaging in general discussions with peers is encouraged, the actual composition and writing of text or code must be done independently. Additionally, avoid copying and pasting text or code from the Internet or generative AI; instead, strive to rephrase content in your own words. If you need assistance, do not hesitate to approach the course staff. All work must be submitted through Blackboard, and late submissions will only be considered if arrangements are made in advance or if extraordinary circumstances warrant an exception.

### Grading Policy

Your grade is individual and is not influenced by the grades of your peers. Our grading system does not involve curving; every student has the potential to achieve an A. The grading process aims to maintain a standard of excellence and provide constructive feedback, rather than comparing students to each other. Your final letter grade will be determined through a combination of factors, including lessons (readings and exercises), homework assignments, programming tasks, and exams. The approximate percentage breakdowns are as follows:

Activities	Percent (%)
Lessons (readings and exercises)	25
Homework assignments	25
Programming assignments	25
Exams	25

Additionally, up to 5% bonus points are available for class attendance and active participation. To earn these extra points, you must attend lectures punctually and engage in class discussions constructively and well-prepared. This includes asking or answering questions that reflect your reading efforts and attempting to comprehend the material. Meeting deadlines for classwork and activities is equally important. Participation will be assessed through Blackboard tracking tools, discussions, and group assignments.

Adherence to due dates is vital. There will be no provision for makeup assignments or late submissions unless a substantial and compelling reason, sanctioned by the course staff, exists. Unless otherwise directed, all coursework should be submitted electronically through Blackboard. Ensure that your work is submitted by the specified due date

or secure special permission from the course staff prior to the deadline. Extensions beyond the subsequent assignment will not be granted, except under exceptional circumstances.

Final letter grades will be determined by computing the percentage of total points you have accumulated. The nominal conversion from percentage score to letter grade is as follows:

Letter grade	Percent (%)	Performance
A	90-100	Excellent
B	80-89	Good
C	70-79	Average
D	60-69	Poor
F	0-59	Failing

The instructor reserves the right to adjust these criteria downward by considering overall class performance, such as designating an A for scores of 88% or higher.

### **Attendance/Participation Policy**

Consistent and punctual class attendance is crucial to your success in this course. Regular attendance is not only expected but mandatory, as it significantly impacts your academic progress. The instructor reserves the right to impose penalties for unexcused absences, including reducing your final grade by one point for each unexcused absence beyond three. The following excerpt is from the 2024-2025 Catalog:

“The student is expected to attend all classes and laboratory sessions, and attendance is mandatory for all freshman-level courses. It is the responsibility of the student to inform each instructor of extended absences. When, in the judgment of the instructor, a student has been absent to such a degree as to impair his or her status relative to credit for the course, the instructor can drop the student from the class with a grade of W before the course drop deadline and with a grade of F after the course drop deadline.”

Adhering to regular attendance aligns with academic expectations and ensures compliance with university policies. This commitment will significantly enhance your learning experience and overall performance on the course. Make sure to inform the course staff of any extended absences to avoid any penalties and to maintain your academic standing.

### **Standards of Conduct**

Maintaining a professional and courteous demeanor is essential and aligns with the guidelines outlined in the Handbook of Operating Procedures: Student Conduct and Discipline. All graded assignments, including readings, exercises, homework, and exams, must be completed independently and distinctly reflect your own work. While discussing general ideas with peers is permissible, presenting material copied or transcribed from external sources—such as individuals, books, websites, or generative AI—as your own work is strictly prohibited.

According to university policy, “Plagiarism means the appropriation, buying, receiving as a gift, or obtaining by any means another’s work and the unacknowledged submission or incorporation of it in one’s academic work offered for credit, or using work in a paper or assignment for which the student had received credit in another course without direct permission of all involved instructors. Plagiarism is a serious violation of university policy and will not be tolerated.”

In adherence to university regulations, any suspected instances of plagiarism will be reported to the Office of Student Conduct and Conflict Resolution (OSCCR) for thorough investigation and appropriate action. Upholding the highest standards of academic integrity is crucial to maintaining the educational environment and upholding the values of this institution.

### **Accommodations**

If you have a disability and need classroom accommodation, we encourage you to contact the Center for Accommodations and Support Services (CASS) promptly. You can reach them at 747-5148 or via email at [cass@utep.edu](mailto:cass@utep.edu). Additionally, you are welcome to visit their office located in UTEP Union East, Room 106. For further information and details, please visit the CASS website: <http://www.sa.utep.edu/cass>. Your proactive communication with CASS ensures that we can provide the necessary support and accommodations to facilitate your academic success.

## Schedule

The table below presents a planned schedule for the course. For the most current schedule, consult the course website.

Dates		Topics	Readings	Assignments
Week 1	Aug. 27, 29	About CS 3360 Preliminaries	Chapter 1	
Week 2	Sep. 3, 5	Describing syntax	Sections 3.1-3.3	Homework 1
Week 3	Sep. 11, 12	Attribute grammar Web scripting with PHP	Section 3.4 E-book	
Week 4	Sep. 17, 19	PHP	E-book	Programming 1
Week 5	Sep. 24, 26	PHP		
Week 6	Oct. 1, 3	Names, bindings, and scopes	Chapter 5	Homework 2
Week 7	Oct. 8, 10	Data types <b>Exam 1</b>	Sections 6.1-6.9	
Week 8	Oct. 15, 17	Object-oriented programming Dart	Sections 12.1-12.6	Homework 3
Week 9	Oct. 22, 24	Dart	E-book	Programming 2
Week 10	Oct. 29, 31	Dart		
Week 11	Nov. 5, 7	Functional programming Haskell	Sections 15.1-15.3 Section 15.8	
Week 12	Nov. 12, 14	Haskell	E-book	Programming 3
Week 13	Nov. 19, 21	Haskell Describing semantics	Section 3.5	
Week 14	Nov. 26, 28	Subprograms <i>Thanksgiving holiday</i>	Sections 9.1-9.6	
Week 15	Dec. 3, 5	Logic programming and Prolog	Chapter 16	
Week 16	Dec. 12	<b>Final</b> at 4:00 pm – 6:45 pm		

## Important Dates

August 26:	Classes begin
September 2:	Labor Day holiday – university closed
September 11:	Census Day
October 10:	Exam 1
November 1:	Course drop/withdrawal deadline
November 28-29:	Thanksgiving holiday - university closed
December 5:	Last day of classes
December 6:	Dead day
December 12:	Final on Thursday at 4:00 pm – 6:45 pm

# CS 3360: Programming Language Concepts

## Learning Outcomes

### Level 1: Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level. Upon successful completion of this course, students will be able to:

- 1a. Describe broad trends in the history of the development of programming languages.
- 1b. Explain the stages of programming language interpretation and compilation.
- 1c. Understand data and control abstractions of programming languages.
- 1d. Understand how the attribute grammars describe static semantics.
- 1e. Describe ways to formally specify the dynamic semantics of small subsets of programming languages, such as expressions and control structures.
- 1f. Understand code snippets written in a paradigm beyond imperative, object-oriented, and functional, e.g., algebraic, aspect-oriented, logic, or probabilistic languages.

### Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

- 2a. Define the syntax of a small context-free grammar in BNF.
- 2b. Define the syntax of a small subset of a programming language in BNF.
- 2c. Compare different approaches to naming, storage bindings, typing, scope, and data types.
- 2d. Analyze design dimensions of subprograms, including parameter passing methods, sub-programs as parameters, and overloaded subprograms.
- 2e. Be able to write programs to solve simple problems in a purely functional language.
- 2f. Be able to write programs to solve simple problems in a scripting language.

### Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the students can apply the material in new situations. This is the highest level of mastery. Upon successful completion of this course, students will be able to:

- 3a. Evaluate modern, representative programming languages critically, considering design concepts and design alternatives, and implementation issues for variables, types, expressions, control structures, and program modules.
- 3b. Choose a suitable programming paradigm and language for a given problem or domain.