

# CS 3360: Programming Language Concepts

## Fall 2023

CRN: 16171

Lecture: TR 1:30 PM - 2:50 PM in CCSB 1.0202

Instructor: Yoonsik Cheon (ycheon@utep.edu); office hours: TR 4:30 PM – 5:20 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 study concepts and examples of programming languages to acquire the tools necessary for critical evaluation and the rapid mastery of programming languages and constructs.

The course aims to strike a balance between theory and hands-on experience. We will survey the constructs and capabilities typically found in modern programming languages, paying attention to design trade-offs and implementation considerations. By gaining an understanding of the range of possibilities likely to be encountered in a language, students will be prepared to quickly learn new languages throughout their careers. By comprehending the implications of design alternatives, students will be better equipped to anticipate the problems likely to arise when using a new language. Furthermore, the presentation of design alternatives and trade-offs lays the groundwork for future advanced studies in compilers and programming language semantics. To exemplify the discussion on general programming language characteristics, we will delve into several languages in greater detail, such as Dart (a modern object-oriented language), Haskell (a functional language), Prolog (a logic-programming language), and PHP (a Web scripting language). Students will gain practical experience with each programming paradigm by completing a programming project in each of the chosen languages.

### Textbooks

The course textbook is *Concepts of Programming Languages* by Robert W. Sebesta, 12th edition, published by Addison Wesley in 2018. The book should be available at the UTEP bookstore, and you are expected to obtain a copy for use in this course, as we will assign reading assignments from the textbook. Additionally, we recommend 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 references are accessible for free through the UTEP Library; you can use a VPN if accessing them from outside the UTEP domain. We will also assign other supplementary readings from the Internet.

### Examinations

There will be both a mid-term exam and a final exam in this course. The mid-term exam will occur during the regular class session and will be 80 minutes in duration, while the final exam is scheduled according to the university's specified date. A makeup exam will only be considered in exceptional or unavoidable situations, such as an incapacitating illness or attendance at a conference for presentation purposes. If you find yourself in such a situation that necessitates a makeup exam, it is crucial to inform the course staff as early as possible. If you are planning to attend a conference for a paper presentation, you must make arrangements for your exam well in advance. Regardless of the circumstances, you must provide an official document explaining your situation before being eligible for a makeup exam.

## Assignments

There will be three types of assignments in this course: *reading*, *homework*, and *programming* (refer to page 4 for details). You should anticipate dedicating approximately 3-4 hours per week to reading and homework assignments, and an average of 3 hours per week to programming tasks. It is important to note that each programming assignment is projected to demand 8-10 hours of effort, potentially leading to a heavier workload in weeks when programming assignments are due (compared to lighter weeks).

- Reading assignments will require you to go through the textbook and prepare for the upcoming week's lessons. These readings will be followed by quizzes. Quizzes are designed to ensure that you have completed the weekly reading task and have grasped the core concepts from recent lessons. Generally, quizzes will take about 10 minutes each and will encompass the material assigned for the upcoming lessons as well as selected content from previous sessions. Please be aware that there will be no opportunities for makeup quizzes, as the answers will be accessible after the designated due dates.
- Homework assignments will involve completing exercises from the textbook's chapters; assignments involving material not covered in lectures will be evaluated generously.
- Programming assignments are designed to provide you with hands-on experience in specific languages and programming paradigms. These include PHP for Web scripting, Dart for contemporary object-oriented programming, Haskell for functional programming, and Prolog for logic programming.

It is mandatory that you complete all assignments individually unless stated otherwise. While you can engage in general discussions with your peers, you should compose and write the text or code on your own. Additionally, refrain from copying and pasting text or code from the Internet; strive to rephrase the content in your own words. Should you require assistance, don't hesitate to approach the instructor or the TA. All your 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 does not depend on the grades of your peers. Our grading system does not involve curving; every student has the potential to achieve an A. The grading process is not aimed at comparing you to others, but rather at maintaining a standard of excellence and providing constructive feedback. Your final letter grade will be determined through a combination of factors, including lessons (quizzes and exercises), homework assignments, programming tasks, and exams. The following are the approximate percentage breakdowns::

Activities	Percent (%)
Lessons (readings, quizzes, exercises, etc.)	25
Homework assignments	25
Programming assignments	25
Exams	25

Up to 5% bonus points are also allocated for class attendance and active participation. To attain these extra points, it is necessary for you to attend lectures punctually and engage in class discussions in a constructive and well-prepared manner. This may involve asking or answering questions that reflect your reading efforts and attempt to comprehend the material. Meeting deadlines for classwork and activities is equally important. We will assess and evaluate your participation in the course by utilizing Blackboard tracking tools, discussions, blogs, chat sessions, and group assignments.

It is vital to closely adhere to due dates. There will be no provision for makeup assignments or tardy submissions unless there exists a substantial and compelling reason, sanctioned by the instructor. Unless otherwise directed, all coursework for this course should be submitted electronically through Blackboard. You are required to ensure that your work is submitted by the stipulated due date or to secure special permission from the instructor prior to the deadline. Unless under exceptional circumstances, extensions beyond the subsequent assignment will not be granted.

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

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**

Regular class attendance is obligatory, as consistent attendance will undoubtedly contribute to your success in the course. The instructor maintains the right to impose penalties for unexcused absences. For instance, your final grade could potentially be reduced by one point for every unexcused absence beyond three. The following passage is extracted from the 2023-2024 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.”

### **Standards of Conduct**

You are expected to uphold a professional and courteous demeanor in accordance with the guidelines outlined in the Handbook of Operating Procedures: Student Conduct and Discipline. All graded assignments, including quizzes, exercises, homework, and exams, must be completed independently and should distinctly reflect your own work. While general discussions with others about your assignments are permissible, you are prohibited from presenting material copied or transcribed from external sources, including individuals, books, or web pages, as your own work.

“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 compliance with university regulations, any suspected instances of plagiarism will be reported to the Office of Student Conduct and Conflict Resolution (OSCCR) for further investigation.

### **Accommodations**

If you have a disability and require classroom accommodations, please get in touch with the Center for Accommodations and Support Services (CASS) at 747-5148, or via email at [cass@utep.edu](mailto:cass@utep.edu). You can also visit their office situated in UTEP Union East, Room 106. For more details, kindly refer to the CASS website: <http://www.sa.utep.edu/cass>.

## 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. 29, 31	About CS 3360 Preliminaries	Chapter 1	
Week 2	Sep. 5, 7	Describing syntax	Sections 3.1-3.3	Homework 1
Week 3	Sep. 12, 14	Attribute grammar Web scripting with PHP	Section 3.4 E-book	
Week 4	Sep. 19, 21	PHP	E-book	Programming 1
Week 5	Sep. 26, 28	PHP		
Week 6	Oct. 3, 5	PHP (lab/demo) Names, bindings, and scopes	Chapter 5	Homework 2
Week 7	Oct. 10, 12	Names, bindings, and scopes Data types	Sections 6.1-6.9	Homework 3
Week 8	Oct. 17, 19	<b>Exam 1</b> Object-oriented programming	Sections 12.1-12.6	
Week 9	Oct. 24, 26	Dart	E-book	
Week 10	Oct. 31, Nov. 2	Dart		Programming 2
Week 11	Nov. 7, 9	Dart (lab/demo) Functional programming	Sections 15.1-15.3	
Week 12	Nov. 14, 16	Introduction to Haskell Haskell	Section 15.8 E-book	
Week 13	Nov. 21, 23	Haskell		Programming 3
Week 14	Nov. 28, 30	Describing semantics Subprograms	Section 3.5 Sections 9.1-9.6	
Week 15	Dec. 5, 7	Logic programming and Prolog	Chapter 16	
Week 16	Dec. 14	<b>Final</b> at 1:00 pm – 3:45 pm		

## Important Dates

August 28:	Classes begin
September 4:	Labor Day holiday – university closed
September 13:	Census Day
October 17:	Exam 1
November 3:	Course drop/withdrawal deadline
November 23-24:	Thanksgiving holiday - university closed
December 7:	Last day of classes
December 8:	Dead day
December 14:	Final on Thursday at 1:00 pm – 3:45 pm

# CS 3360: Design and Implementation of Programming Languages

## 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.