CS 3360: Programming Language Concepts
Spring 2024

CRN: 23910
Lecture: MW 1:30 PM - 2:50 PM in CCSB 1.0202
Instructor: Yoonsik Cheon (ycheon@utep.edu); office hours: MW 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 explore programming languages, focusing on both theoretical foundations and practical applications. Our primary goal is to equip you with the essential tools for critically evaluating and rapidly mastering diverse programming languages and constructs. The course maintains a balance between theoretical underpinnings and hands-on experiences. We will survey the various constructs and capabilities prevalent in modern programming languages, examining design trade-offs and implementation considerations. By gaining a deep understanding of the spectrum of possibilities encountered in languages, you will be well-prepared to swiftly adapt to new programming languages throughout your professional journey. Understanding the implications of design alternatives is key to navigating potential challenges when working with a new language. Moreover, the exploration of design alternatives and trade-offs lays a solid foundation for future advanced studies in compilers and programming language semantics. To exemplify discussions on general programming language characteristics, we will delve into several languages in greater detail. These include Dart (a modern object-oriented language), Haskell (a functional language), PHP (a Web scripting language), and Prolog (a logic-programming language). As part of your practical learning experience, you will undertake a programming project in each of these chosen languages, providing hands-on exposure to different programming paradigms.

Textbooks

Our core textbook is *Concepts of Programming Languages* by Robert W. Sebesta, 12th edition, published by Addison Wesley in 2018. You can find this book at the UTEP bookstore, and it is crucial to acquire a copy for use in the course, as reading assignments will be assigned from this text. 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 references are available for free through the UTEP Library, accessible via VPN if you are outside the UTEP domain. In addition to these texts, we will assign other supplementary readings from the Internet.

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 (please refer to page 4 for detailed information). 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, compared to lighter weeks.

- Reading assignments: These tasks involve delving into the textbook to prepare for upcoming lessons. Following these readings, quizzes will be administered. The purpose of quizzes is to ensure 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 be aware 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, and assignments involving material not covered in lectures will be generously evaluated.
- Programming assignments: These assignments are to provide hands-on experience in specific languages and programming paradigms, covering PHP for Web scripting, Dart for contemporary object-oriented programming, and Haskell for functional programming.

It is imperative to complete all assignments individually unless explicitly stated otherwise. While engaging in general discussions with peers is encouraged, the composition and writing of text or code should be done independently. Additionally, refrain from copying and pasting text or code from the Internet; strive to rephrase content in your own words. If assistance is needed, do not hesitate to approach the instructor or TA. 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 not to compare you to others but to maintain a standard of excellence and provide 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 approximate percentage breakdowns are as follows:

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 earn 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 attempting 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, 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 Policy

Consistent and punctual class attendance is a crucial factor contributing to your success in this course. Regular attendance is not only expected but mandatory, as it plays a pivotal role in your academic progress. The instructor holds the right to impose penalties for unexcused absences. Specifically, your final grade could potentially be reduced by one point for each unexcused absence beyond three. The following excerpt is derived 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."

In essence, adhering to regular attendance not only aligns with academic expectations but also ensures compliance with university policies. This commitment will significantly enhance your learning experience and overall performance on the course.

Standards of Conduct

Your adherence to a professional and courteous demeanor is expected in alignment with the guidelines articulated in the Handbook of Operating Procedures: Student Conduct and Discipline. All graded assignments, encompassing quizzes, exercises, homework, and exams, must be completed independently, distinctly reflecting your own work. While engaging in general discussions with others about your assignments is permissible, presenting material copied or transcribed from external sources, including individuals, books, or web pages, 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 compliance with 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 essential to the educational environment and 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. 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 outlines the anticipated schedule for the course. For the latest updates and adjustments, please refer to the course website for real-time information.

Dates		Topics	Readings	Assignments
Week 1	Jan. 17	About CS 3360	Chapter 1	
Week 2	Jan. 22, 24	Preliminaries Describing syntax	Sections 3.1-3.3	Homework 1
Week 3	Jan. 29, 31	Describing syntax Attribute grammar	Section 3.4	
Week 4	Feb. 5, 7	Web scripting with PHP	E-book	Programming 1
Week 5	Feb. 12, 14	PHP		
Week 6	Feb. 19, 21	PHP Names, bindings, and scopes	Chapter 5	Homework 2
Week 7	Feb. 26, 28	Names, bindings, and scopes Data types	Sections 6.1-6.9	Homework 3
Week 8	Mar. 4, 6	**Exam 1** Object-oriented programming	Sections 12.1-12.6	
Week 9	Mar. 11, 13	Spring break (no classes)		
Week 10	Mar. 18, 20	Dart	E-book	
Week 11	Mar. 25, 27	Dart		Programming 2
Week 12	Apr. 1, 3	Dart Functional programming	Sections 15.1-15.3	
Week 13	Apr. 8, 10	Introduction to Haskell Haskell	Section 15.8 E-book	
Week 14	Apr. 15, 17	Haskell		Programming 3
Week 15	Apr. 22, 24	Describing semantics Subprograms	Section 3.5 Sections 9.1-9.6	
Week 16	Apr 29, May 1	Logic programming and Prolog	Chapter 16	
Week 17	May 8	**Final** at 4:00 PM – 6:45 PM		

Important Dates

January 15: Dr. Martin Luther King, Jr. holiday – university closed
January 16: Classes begin
January 31: Census Day
March 4: Exam 1
March 11-15: Spring break
March 28: Course drop/withdrawal deadline
March 29: Cesar Chavez holiday - no classes; Spring study day
May 2: Last day of classes
May 3: Dead day
May 8: Final on Wednesday at 4:00 PM - 6:45 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.