CS 3360: Design and Implementation of Programming Languages
Spring 2023

CRN: 26272
Lecture: MW 1:30 PM - 2:50 PM in LART 318
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 study concepts and examples of programming languages to acquire the tools necessary for critical evaluation and rapid mastery of programming languages and constructs.

The course attempts to balance theory and hands-on experience. We will survey the constructs and capabilities typically found in modern programming languages with 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 learn new languages quickly throughout their careers. By understanding the implications of design alternatives, students will be better able to anticipate the problems likely to arise in using a new language. Also, the presentation of design alternatives and trade-offs lays the groundwork for future advanced study of compilers and programming language semantics. To instantiate the discussion of general programming language characteristics, several languages will be presented in more detail: e.g., 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 Robert W. Sebesta, *Concepts of Programming Languages*, 12th edition, Addison Wesley, 2018. The book should be available at the UTEP bookstore, and you should acquire a copy for your use in this course, as we will take reading assignments from the textbook. In addition, recommended are the following supplementary books.

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

Electronic books of the recommended references are accessible for free through the UTEP Library; use VPN from outside the UTEP domain. We will take other supplemental readings from the Internet.

Examinations

There will be one mid-term exam and the final exam. The mid-term exam will take place during the regular class session and will be 80 minutes long, and the final exam will take place on the date specified by the university. We grant a makeup exam only in an unusual or unavoidable circumstance, e.g., an incapacitating illness or a presentation at a conference. If you are in this circumstance that may warrant a makeup exam, you must notify the course staff as early as possible. If you plan to attend a conference to present a paper, you must arrange your exam *in advance*. In all circumstances, you must provide an official document explaining your situation before your make-up exam.

Assignments

There will be three kinds of assignments: *reading*, *homework*, and *programming* (see page 4). You should expect to spend about 3-4 hours per week on reading and homework assignments and an average of 3 hours per week on

programming assignments. Note each programming assignment is estimated to require 8-10 hours, so your workload in weeks that programming assignments are due may be higher (with other weeks being correspondingly lower).

- Reading assignments ask you to read the textbook and prepare for the coming week's lessons. There will be quizzes on your readings. A quiz aims to ensure that you have done the weekly reading assignment and to verify that you have mastered the main concepts of recent lessons. Quizzes typically will require about 10 minutes long and will cover the material you read for the upcoming lessons plus selected ones from previous lessons. There will be no make-up for missed quizzes, as the answers will be available after the due dates..
- Homework assignments ask you to do the chapter exercises taken from the textbook; problems that use material not covered in lectures will be graded generously.
- Programming assignments are designed to allow you to gain hands-on experience with specific languages and programming paradigms (PHP for Web scripting, Dart for modern object-oriented programming, Haskell for functional programming, and Prolog for logic programming).

You must do all your assignments individually unless specified otherwise; refer to the handouts for possible pair work. It means that while you may discuss them in general terms with your classmates, you should compose or write text or code alone. Also, do not copy and paste text or code from the Internet; write them in your own words. If you need help, see the instructor or the TA. You must submit all your work through Blackboard, and *no late submission will be accepted* unless arrangements are made in advance or unless unusual circumstances warrant an exception.

Grading Policy

Your grade is independent of everyone else's grade. We do not grade on a curve; everyone can earn an A. The purpose of grading is not to rank you but to uphold a standard of quality and give feedback. We will calculate your final letter grade using a combination of lessons (quizzes and exercises), homework assignments, programming assignments, and exams. Below are the approximate percentages:

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

There are also up to 5% bonus points for class attendance and participation. To earn this bonus, you must arrive at lectures on time and participate in class discussions in a constructive and prepared manner, e.g., by asking or answering questions that demonstrate that you have read and attempted to understand the material. You should also complete classwork and activities on time. We will monitor, track, and score your participation in the course partly using Blackboard tracking tools, discussions, blogs, chat sessions, and group work.

Be sure to pay close attention to due dates -- there will be no makeup assignments or late work accepted without a serious and compelling reason and the instructor's approval. You should submit all your work for this course electronically through Blackboard unless otherwise instructed. You must submit your work by the due date or obtain special permission from the instructor before the due date. Except under extreme circumstances, you won't get an extension beyond the next assignment.

We will calculate your final letter grades based on the percentage of your total points earned. Below is the nominal percentage-score-to-letter-grade conversion.

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, e.g., 88% or higher represents an A.

Attendance/Participation

Class attendance is required; you should understand that your success in the course will improve by attending classes regularly. *The instructor reserves the right to penalize unexcused absences, e.g., your final grade may be lowered by one point for each unexcused absence above three.* The following is excerpted from the 2022-2023 Catalog.

"The student is expected to attend all classes and laboratory sessions and attendance are 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 should conduct yourself professionally and courteously as described by the Handbook of Operating Procedures: Student Conduct and Discipline. All graded work (quizzes, exercises, homework, exams) is to be completed independently and should be unmistakably your work, although you may discuss your work with others in a general way. You may not represent the material transcribed or copied from another source, including persons, books, or Web pages, as your 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."

As required by the university, we will report all cases of suspected plagiarism to the Office of Student Conduct and Conflict Resolution (OSCCR) for further review.

Accommodations

If you have a disability and need classroom accommodations, contact the Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at http://www.sa.utep.edu/cass.

Schedule

The following table shows a planned schedule for the course; refer to the course website for an up-to-date schedule.

Dates		Topics	Readings	Assignments
Week 1	Jan. 18	About CS 3360	Chapter 1	
Week 2	Jan. 23, 25	Preliminaries Describing syntax	Sections 3.1-3.3	Homework 1
Week 3	Jan. 30, Feb. 1	Describing syntax Attribute grammar	Section 3.4	
Week 4	Feb. 6, 8	Web scripting with PHP	E-book	Programming 1
Week 5	Feb. 13, 15	PHP		
Week 6	Feb. 20, 22	PHP Names, bindings, and scopes	Chapter 5	Homework 2
Week 7	Feb. 27, Mar. 1	Names, bindings, and scopes Data types	Sections 6.1-6.9	Homework 3
Week 8	Mar. 6, 8	**Exam 1** Object-oriented programming	Sections 12.1-12.6	
Week 9	Mar. 13, 15	Spring break (no classes)		
Week 10	Mar. 20, 22	Dart	E-book	
Week 11	Mar. 27, 29	Dart		Programming 2
Week 12	Apr. 3, 5	Dart Functional programming	Sections 15.1-15.3	
Week 13	Apr. 10, 12	Introduction to Haskell Haskell	Section 15.8 E-book	
Week 14	Apr. 17, 19	Haskell		Programming 3
Week 15	Apr. 24, 26	Describing semantics Subprograms	Section 3.5 Sections 9.1-9.6	
Week 16	May 1, 3	Logic programming and Prolog	Chapter 16	
Week 17	May 10	**Final** at 4:00 PM – 6:45 PM		

Important Dates

January 16:	Dr. Martin Luther King, Jr. holiday – university closed
January 17:	Classes begin
February 1:	Census Day
March 6:	Exam 1
March 13-17:	Spring break
March 30:	Course drop/withdrawal deadline
March 31:	Cesar Chavez holiday - no classes
April 5:	Course drop/withdrawal deadline
April 7:	Spring study day
May 4:	Last day of classes
May 5:	Dead day
May 10:	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 using 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.