

CS 3360: Design and Implementation of Programming Languages

Fall 2022

CRN: 11660

Lecture: MW 1:30 PM - 2:50 PM in EDUC 313

Instructor: Yoonsik Cheon (ycheon@utep.edu); office hours: MW 3:00 PM – 4:20 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 with the goal of acquiring 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 textbook should be available at the UTEP bookstore, and you are expected to acquire a copy for your use in this course, as reading assignments will be taken from the textbook. In addition, the following supplementary textbooks are recommended:

Kevin Tatroe and Peter MacIntyre, *Programming PHP: Creating Dynamic Web Pages*, 4th 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 available for free through UTEP Library from UTEP domain; use VPN from outside UTEP domain. Other supplemental readings will be taken from the Internet.

Examinations

There will be two exams: *mid-term* and *final*. The final exam may be comprehensive. The mid-term exams will take place during the regular class session and will be 80 minutes in length, and the final exam will take place on the date specified by the university (see page 4 for tentative exam dates). Make-up exams will be given only when you have unusual circumstances, such as incapacitating illness or presenting a research paper at a conference. If you believe that you have an unusual circumstance that warrants a make-up exam, notify us as soon as possible. If you will be attending a conference or other event, you must make arrangements for a make-up exam *in advance*. Under all circumstances, you are required to provide official documentation before a make-up will be administered.

Assignments

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

of 3 hours per week for programming assignments. Note, however, that 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 to read the textbook and prepare for the coming week's lessons. There will be quizzes on your readings. The purpose of a quiz is to ensure that you have done the weekly reading assignment and to verify that you have mastered the major concepts of recent lessons. Quizzes typically will be about 10 minutes in length and will cover the material assigned to be read for the upcoming lessons plus selected concepts from previous lessons. There will be no make-up on missed quizzes.
- Written homework assignments ask you to do exercises from the textbook; exercises that use material not covered in lectures will be graded generously.
- Programming assignments are designed to allow you to gain hands-on experiences 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).

All assignments shall be submitted through Blackboard. For all assignments, no late submission will be accepted unless arrangements have been made in advance or unless unusual circumstances warrant an exception. Unless otherwise specified, all assignments are to be done individually. While you may discuss the assignment in general terms with others, your solutions should be composed, designed, written, and tested by yourself alone. If you need help, consult the TA or the instructor.

Grading Policy

Your grade is independent of anyone else's grade. We do not grade on a curve, and everyone can earn an A in this course. The purpose of grading is not to rank you, but to uphold a standard of quality and to give you feedback. Your final letter grade will be calculated based on a combination of lessons, homework assignments, programming assignments, and exams. The approximate percentages are shown below:

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 deadlines—there will be no makeup assignments or quizzes, or late work accepted without a serious and compelling reason and instructor approval. All work and assignments for this course will be submitted electronically through Blackboard unless otherwise instructed. They must be submitted by the given deadline or special permission must be requested from the instructor before the due date. Extensions will not be given beyond the next assignment except under extreme circumstances.

Final letter grades assigned for this course will be based on the percentage of total points earned and are assigned as follows. The nominal percentage-score-to-letter-grade conversion 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, e.g., so that 88% or higher represents an A, based on overall class performance. The criteria will not be adjusted upward, however.

Attendance/Participation

Class attendance is required; you should understand that your success in the course will improve greatly 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.

You should understand that your success in the course will improve greatly by participating/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 2021-2022 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 conduct yourself in a professional and courteous manner, as prescribed by the Handbook of Operating Procedures: Student Conduct and Discipline. All graded work (homework, projects, and exams) is to be completed independently and should be unmistakably your own work, although you may discuss your work with others in a general way. You may not represent as your own work material that is transcribed or copied from another source, including persons, books, or Web pages. “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 own 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. All cases of suspected plagiarism will be reported to the Office of Student Conduct and Conflict Resolution (OSCCR) for further review.

Accommodations

If you have a disability and need classroom accommodations, please 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>.

COVID-19 Precautions

Please stay home if you have been diagnosed with COVID-19 or are experiencing COVID-19 symptoms. If you are feeling unwell, please let me know as soon as possible, so that we can work on appropriate accommodations. If you have tested positive for COVID-19, you are encouraged to report your results to covidaction@utep.edu, so that the Dean of Students Office can provide you with support and help with communication with your professors. The Student Health Center is equipped to provide COVID-19 testing.

The Center for Disease Control and Prevention recommends that people in areas of substantial or high COVID-19 transmission wear face masks when indoors in groups of people. The best way that Miners can take care of Miners is to get the vaccine. If you still need the vaccine, it is widely available in the El Paso area and will be available at no charge on campus during the first week of classes. For more information about the current rates, testing, and vaccinations, please visit <http://epstrong.org>.

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	Aug. 22, 24	About CS 3360 Preliminaries	Chapter 1	
Week 2	Aug. 29, 31	Describing syntax	Sections 3.1-3.3	Homework 1
Week 3	Sep. 5, 7	<i>Labor Day – no class</i> Attribute grammar	Section 3.4	
Week 4	Sep. 12, 14	Web scripting with PHP	E-book	Programming 1
Week 5	Sep. 19, 21	PHP		
Week 6	Sep. 26, 28	PHP Names, bindings, and scopes	Chapter 5	Homework 2
Week 7	Oct. 3, 5	Names, bindings, and scopes Data types	Sections 6.1-6.9	Homework 3
Week 8	Oct. 10, 12	Object-oriented programming Exam 1	Sections 12.1-12.6	
Week 9	Oct. 17, 19	Dart	E-book	
Week 10	Oct. 24, 26	Dart		Programming 2
Week 11	Oct. 31, Nov. 2	Dart Function programming	Sections 15.1-15.3	
Week 12	Nov. 7, 9	Introduction to Haskell Haskell	Section 15.8 E-book	
Week 13	Nov. 14, 16	Haskell		Programming 3
Week 14	Nov. 21, 23	Describing semantics Subprograms	Section 3.5 Sections 9.1-9.6	
Week 15	Nov. 28, 30	Logic programming and Prolog	Chapter 16	
Week 16	Dec. 7	Final at 4:00 pm – 6:45 pm		

Important Dates

August 22:	Classes begin
September 5:	Labor Day holiday – university closed
September 7:	Census Day
October 12:	Exam 1
October 28:	Course drop/withdrawal deadline
November 24-25:	Thanksgiving holiday - university closed
December 1:	Last day of classes
December 2:	Dead day
December 7:	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 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 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 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 with respect to design concepts, design alternatives, and implementation considerations for variables, types, expressions, control structures, and program modules.
- 3b. Choose a suitable programming paradigm and language for a given problem or domain.