CS 3360: Design and Implementation of Programming Languages
Spring 2021

CRN: 21684
Live session: MW 12:00 PM - 1:20 PM on Blackboard
Instructor: Yoonsik Cheon (ycheon@utep.edu); office hours: MW 1:30 PM - 2:30 PM on MS Teams
Teaching assistant: Ivan Gastelum (igastelum@miners.utep.edu); office hours: 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 on 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 (see course website). Other supplemental readings will be taken from the Internet.

Course Structure

This course is entirely online and delivered through the Blackboard Learn virtual learning environment. You will use your UTEP account to login to the course from the Blackboard link in the My UTEP page (https://my.utep.edu).

In Blackboard Learn, you will access online lessons, course materials, and resources. You will participate in a blend of self-paced and group-paced activities using Blackboard Learn and alternative Internet-based technologies. There will be learning modules containing tasks and activities such as readings, exercises, coding, discussions, quizzes, blogs, and presentations/demos. You will need to complete these semi-weekly lessons, or learning modules (see Figure 1 below). The asynchronous nature of most tasks and activities will permit you to complete them, or take part, at times most convenient to you.

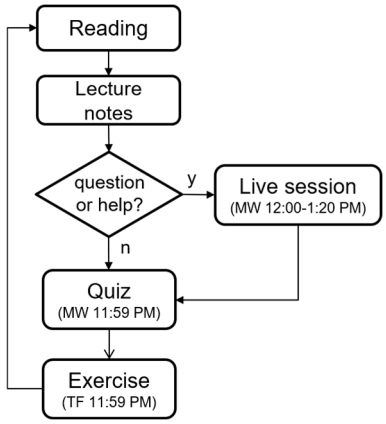

Figure 1. Complete semi-weekly lessons

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*. All assignments will be announced through Blackboard (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 work load 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 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).

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, semester project, and exam. The approximate percentages are shown below:

Activities	Percent (%)
Lessons (readings, quizzes, exercises, etc.)	30
Homework assignments	20
Programming assignments	25
Exams	25

There are also up to 5% bonus points for class attendance and participation; attendance is determined by participation in the learning activities of the course. To earn this bonus, you must participate in most online activities as listed in the course content and complete them on time. We will monitor, track, and score your participation in the course 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 Learn unless otherwise instructed. They must be submitted by the given deadline or special permission must be requested from 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

Attendance in the course is determined by participation in the learning activities of the course. Students are expected to participate in all online activities as listed in the course content. The asynchronous nature of most activities will permit you to take part at times most convenient to you.

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 2020-2021 Catalog.

"The student is expected to attend all classes and laboratory sessions. 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, 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 Dean of Students 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 www.sa.utep.edu/cass.

COVID-19 Precautions

You must STAY AT HOME and REPORT if you (1) have been diagnosed with COVID-19, (2) are experiencing COVID-19 symptoms, or (3) have had recent contact with a person who has received a positive coronavirus test. Reports should be made at screening.utep.edu. If you know of anyone who should report any of these three criteria, you should encourage them to report. If the individual cannot report, you can report on their behalf by sending an email to COVIDaction@utep.edu.

For each day that you attend campus—for any reason—you must complete the questions on the UTEP screening website (screening.utep.edu) prior to arriving on campus. The website will verify if you are permitted to come to campus. Under no circumstances should anyone come to class when feeling ill or exhibiting any of the known COVID-19 symptoms. If you are feeling unwell, please let me know as soon as possible, and alternative instruction will be provided. Students are advised to minimize the number of encounters with others to avoid infection.

Wear face coverings when in common areas of campus or when others are present. You must wear a face covering over your nose and mouth at all times in this class. If you choose not to wear a face covering, you may not enter the classroom. If you remove your face covering, you will be asked to put it on or leave the classroom. Students who refuse to wear a face covering and follow preventive COVID-19 guidelines will be dismissed from the class and will be subject to disciplinary action according to Section 1.2.3 *Health and Safety* and Section 1.2.2.5 *Disruptions* in the UTEP Handbook of Operating Procedures.

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. 20	About CS 3360		
Week 2	Jan. 25, 27	Preliminaries Describing syntax	Chapter 1 Sections 3.1-3.3	Homework 1
Week 3	Feb. 1, 3	Describing syntax Attribute grammar	Section 3.4	
Week 4	Feb. 8, 10	Web scripting with PHP	E-book	Programming 1
Week 5	Feb. 15, 17	PHP		
Week 6	Feb. 22, 24	Lab/demo (PHP) Names, bindings, and scopes	Chapter 5	Homework 2
Week 7	Mar. 1, 3	Names, bindings, and scopes Data types	Sections 6.1-6.9	
Week 8	Mar. 8, 10	**Exam 1** Object-oriented programming	Sections 12.1-12.6	Homework 3
Week 9	Mar. 15, 17	Spring break		
Week 10	Mar. 22, 24	Dart	E-book	
Week 11	Mar. 29, 31	Dart		Programming 2
Week 12	Apr. 5, 7	Lab/demo (Dart) Function programming	Sections 15.1-15.3	
Week 13	Apr. 12, 14	Introduction to Haskell Haskell	Section 15.8 E-book	
Week 14	Apr. 19, 21	Haskell Lab/demo (Haskell)		Programming 3
Week 15	Apr. 26, 28	Describing semantics Subprograms	Section 3.5 Sections 9.1-9.6	
Week 16	May 3, 5	Logic programming and Prolog	Chapter 16	
Week 17	May 14	**Final** at 1:00 PM - 3:45 PM		

Important Dates

January 18: Dr. Martin Luther King, Jr. holiday – university closed
January 19: Classes begin
February 3: Census day
March 8: Exam 1
March 15-19: Spring break
March 26: Cesar Chavez holiday – no classes
April 1: Drop/withdrawal deadline
April 2: Study day
May 6: Last day of classes
May 7: Dead day
May 14: Final on Friday 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 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.

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 and EBNF.
2b. Define the syntax of a small subset of a programming language using BNF or EBNF.
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 overload 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 trade-offs, and implementation considerations for scope, binding, data types, expressions, control structures, subprograms, abstract data types, objects, concurrency structures, and exception handling.
3b. 3b. Choose a suitable programming paradigm and language for a given problem or domain.