CS 3360 Design and Implementation of Programming Languages
Summer 2017 Syllabus

Class Time: Monday through Friday 11:40 am - 1:30 pm, BUSN 313

Instructor: Vladik Kreinovich, email vladik@utep.edu, office CCSB 3.0404, office phone (915) 747-6951.

- The instructor's office hours are:
  - Monday 9:30-10:30 am
  - Tuesday 10:30-11:30 am and 1:30-2:30 pm
  - Wednesday 9:30-10:30 am
  - Thursday 9:30-10:30 am and 1:30-2:30 pm
- If you want to come during office hours, there is no need to schedule an appointment.
- If you cannot come during the instructor's office hours, please schedule an appointment in the following way:
  - use the instructor's appointments page http://www.cs.utep.edu/vladik/appointments.html to find the time when the instructor is not busy (i.e., when he has no other appointments), and
  - send him an email, to vladik@utep.edu, indicating the day and time that you would like to meet.

  He will then send a reply email, usually confirming that he is available at this time, and he will place the meeting with you on his schedule.

Course Objectives: Design features of modern programming languages including flow control mechanism and data structures; techniques for implementation of these features.

The prerequisite of this course is CS 2302 with a grade of C or better.
It is also recommended to take:

- CS 3331 (Advanced Object-Oriented Programming) and
- CS 3432 (Computer Architecture I)

before you take this course.

Major Topics Covered in the Course

- Types of programming languages
- A brief history of programming languages
- How to describe syntax
- Lexical analysis and parsing
- Variables: names, bindings, scopes, data types
- Assignment and control statements
- Subprograms
- Object-oriented languages: how to implement abstract data types and encapsulation
- Concurrency
- Exception handling
- Functional programming languages

- Logic 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:

1.1. Explain the broad outlines and major names in the history of development of programming languages

1.2. Recognize major programming languages

1.3. Explain the stages of programming language interpretation and compilation

1.4. Understand context-free grammars and (E)BNF notations

1.5. Explain operational semantics, axiomatic semantics, and denotational semantics as methods of expressing programming language semantics

1.6. Explain 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 in modern programming 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:

2.1. Define syntax of a small context-free grammar in BNF and EBNF

2.2. Use attribute grammars to describe the static semantics of small programming languages

2.3. Define dynamic semantics of small subsets of programming languages, e.g., expressions and control structures

2.4. Evaluate different approaches to naming, storage bindings, typing, scope, and data types

2.5. Select and apply appropriate expressions and control structures for a given programming task

2.6. Analyze design dimensions of subprograms, including parameter passing methods, subprograms as parameters, and overload subprograms

2.7. Analyze and evaluate data and control abstractions of programming languages

Level 3: Synthesis and Evaluation
Level 3 outcomes are those in which the student 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:

3.1. Evaluate modern, representative programming languages critically

3.2. Choose a suitable programming paradigm and language for a given problem or domain

3.3. Design a small, domain-specific programming language, e.g., a test script language, by defining its formal syntax and semantics

Textbook: *Concepts of Programming Languages* by Robert W. Sebesta (11th edition). This book is available at the bookstore and through major online book retailers, and you are expected to acquire a copy for your use in this course. Photocopied textbooks are illegal and their use will not be tolerated.

Assignments: Reading and homework assignments will be handed out or announced in class and in labs. If you miss a class, it is your responsibility to find out what you missed. You should expect to spend at least 10 hours/week outside of class on reading and homework.

Exams and Grading: There will be two tests and a final exam. UTEP has scheduled our final exam on Friday August 4, 1-3:45 pm. Each topic means home assignments (mainly on the sheets of paper, but some on the real computer). Maximum number of points:

- first test: 20
- second test: 20
- home assignments and quizzes: 20
- final exam: 40

The purpose of the exams is to allow you to demonstrate mastery of course concepts. Make-up exams will be given only in extremely unusual circumstances. If you must miss an exam, please meet with an instructor, BEFORE the exam if at all possible.

The nominal percentage-score-to-letter-grade conversion is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- below 60% is an F

We reserve 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.

Homework Assignments: Homework and lab assignments are designed to allow you to practice using the concepts presented in lecture and in your reading. Homework and lab assignments may include written problems, tutorial exercises, and programming problems. Assignments usually will be due at the start of the next class. Late homework will be accepted only in unusual circumstances, by prior arrangement if at all possible.

Homework must be done individually. While you may discuss the problem in general terms with other people, your answers and your code should be written and tested by you alone. If you need help, consult a TA or a professors

Quizzes: The purpose of a quiz is to ensure that you have read the weekly reading assignment and to verify that you have mastered the major concepts of recent lectures. Quizzes typically will be about

5-10 minutes in length and will cover the material assigned to be read for the upcoming lecture plus selected concepts from previous lectures. There will be no make-up on missed quizzes.

Standards of Conduct: You are expected to conduct yourself in a professional and courteous manner, as prescribed by the UTEP Standards of Conduct.

Graded work, e.g., homework and tests, is to be completed independently and should be unmistakably your own work (or, in the case of group work, your team's work), although you may discuss your project with other students in a general way. You may not represent as your own work material that is transcribed or copied from another person, book, or any other source, e.g., a web page.

Academic dishonesty includes but is not limited to cheating, plagiarism and collusion.

- Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying data (for example program outputs) in laboratory reports.
- Plagiarism occurs when someone represents the work or ideas of another person as his/her own.
- Collusion involves collaborating with another person to commit an academically dishonest act.

Professors are required to -- and will -- report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

Disabilities: If you feel you may have a disability that requires accommodation, contact the The Center for Accommodations and Support Services (CASS) at 747-5148, go to Room 106 E. Union, or e-mail to cass@utep.edu.