



CS 326: Programming Language Concepts

General Information:

Term: 2021 Summer Session
Instructor: Staff
Language of Instruction: English
Classroom: TBA
Office Hours: TBA
Class Sessions Per Week: 5
Total Weeks: 5
Total Class Sessions: 25
Class Session Length (minutes): 145
Credit Hours: 4

Course Description:

This course introduces the fundamental concepts on which programming languages are based and the execution models supporting them. Topics includes the following: Language categories, Language design, Programming languages evolution, Syntax, Semantics, Lexical and syntax analyzers, Names, Bindings, Type checking, Scoping, Data types, Expressions, Statements, Statement-level control structures, Subprograms and its implementation, Abstract data types, Functional programming languages, Logic programming languages.

Prerequisite: CS 213 Data Structure or other equivalent course.

Learning Objectives:

After having followed the course activities the student will be able to:

1. Have a good understanding on fundamental concepts of all programming languages.
2. Understand and practice paradigms of programming languages.
3. Understand how the concepts are handled in major programming languages
4. Be able to use language processors for language design and implementation.

Course Materials:

Concepts of Programming Languages, 11th Edition, by Robert W. Sebesta (Author)



Publisher: Pearson; 11 edition (February 16, 2015)

Language: English

ISBN-10: 013394302X

ISBN-13: 978-0133943023

Course Format and Requirements:

The course will take place in a computer lab and the course format including lecture, programming project, and in-class discussion.

The specific topics that will be covered in the classes are listed in the course syllabus. The class period will consist of an active learning environment. During a majority of the class time, students will be actively working on problems under the instructor's guides.

Attendance:

Attendance will not be taken but is strongly recommended. Each student will have three allowed absences and no grade deduction will be made for the first three absences. More than three unexcused absences will result in an automatic reduction in your participation grade, for instance from A- to B+. Your active participation in the class is expected and encouraged.

Course Assignments:

Quizzes:

There will be seven quizzes administered through the whole semester. The quizzes aims to reinforce the conceptual material from the textbook and quizzes problems will be similar to textbook topics and examples on slides. Quizzes will always be completed in the beginning of class. There will be no make-up quizzes. The lowest one will be dropped.

Weekly Programming Projects

The weekly hands-on programming projects are based on course need. These programming projects are all individual work. Students are encouraged to exchange and discuss programming knowledge and ideas together. But each student shall submit completed individual work.

The projects aim to enrich students' knowledge on application of learned concepts, programming languages and programming paradigms. The score will be given based on the correctness of the program. It will count for 30% of your grade for the course.

Exams:



There will be one midterm exam and one cumulative final exam. Both exams will be in open-book format.

No excuse will be accepted if students do not have legitimate excuses for absence in exams. Physician Statement is required for missing the exam due.

Course Assessment:

Class participation	10%
Quizzes	15%
Weekly Programming Project	25%
Midterm exam	20%
Final exam	30%
Total	100%

Grading Scale (percentage):

A+: 98%-100%

A: 93%-97%

A-: 90%-92%

B+: 88%-89%

B: 83%-87%

B-: 80%-82%

C+: 78%-79%

C: 73%-77%

C-: 70%-72%

D+: 68%-69%

D: 63%-67%

D-: 60%-62%

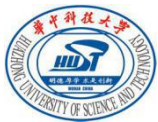
F: Below 60%

Course Schedule:

Week	Topics	Assignments
Week One (Class 1~5)	<ul style="list-style-type: none"> ● Introduction and History of Programming Languages Chapter 1 : Preliminaries	<ul style="list-style-type: none"> • Quiz 1 • Quiz 2 • Textbook Reading • Weekly Programming



	<p>Chapter 2 : Evolution of the Major Programming Languages</p> <ul style="list-style-type: none"> ● Describing Syntax and Semantics (Regular Expressions & Context Free Grammars) <p>Chapter 3 : Describing Syntax and Semantics</p> <ul style="list-style-type: none"> ● Lexical and Syntax Analysis <p>Chapter 4 : Lexical and Syntax Analysis</p>	project
Week Two (Class 6~10)	<ul style="list-style-type: none"> ● Names, Bindings and Types of Data <p>Chapter 5 : Names, Bindings, Type Checking, and Scopes</p> <p>Chapter 6 : Data Types</p>	<ul style="list-style-type: none"> • Quiz 3 • Textbook Reading • Weekly Programming project
Week Three (Class 11~15)	<ul style="list-style-type: none"> ● Expression, Assignments Statement & Flow Control <p>Chapter 7 : Expressions and Assignment Statements</p> <p>Chapter 8 : Statement-Level Control Structures</p> <ul style="list-style-type: none"> ● Subprograms and Implementation of Subgroups <p>Chapter 9 : Subprograms</p> <p>Chapter 10 : Implementing Subprograms</p>	<ul style="list-style-type: none"> • Quiz 4 • Textbook Reading • Weekly Programming project • Review • Midterm exam
Week Four (Class 16~20)	<ul style="list-style-type: none"> ● Advanced Language Concepts <p>Chapter 13 : Concurrency</p> <p>Chapter 14 : Exception Handling and Event Handling</p> <ul style="list-style-type: none"> ● Object Oriented Programming Languages <p>Chapter 11 : Abstract Data Types and Encapsulation Constructs</p>	<ul style="list-style-type: none"> • Quiz 5 and Quiz 6 • Textbook Reading • Weekly Programming project
Week Five (Class 21~25)	<ul style="list-style-type: none"> ● Object Oriented Programming Languages <p>Chapter 12 : Support for Object Oriented Programming</p> <ul style="list-style-type: none"> ● Functional Programming Languages <p>Chapter 15 : Functional Programming Languages</p> <ul style="list-style-type: none"> ● Logic Programming Languages <p>Chapter 16 : Logic Languages</p>	<ul style="list-style-type: none"> • Quiz 7 • Textbook Reading • Weekly Programming project • Review • Final exam



Note: This is a tentative course schedule. Some sections are subjected to change or be omitted according to the instructors' final arrangement.

Academic Integrity:

Students are encouraged to study together, and to discuss lecture topics with one another, but all other work should be completed independently.

Students are expected to adhere to the standards of academic honesty and integrity that are described in the Huazhong University of Science & Technology's *Academic Conduct Code*. Any work suspected of violating the standards of the *Academic Conduct Code* will be reported to the Dean's Office. Penalties for violating the *Academic Conduct Code* may include dismissal from the program. All students have an individual responsibility to know and understand the provisions of the *Academic Conduct Code*.

Special Needs or Assistance:

Please contact the Administrative Office immediately if you have a learning disability, a medical issue, or any other type of problem that prevents professors from seeing you have learned the course material. Our goal is to help you learn, not to penalize you for issues which mask your learning.