

CMSE 318 – CMPE 410 Principles of Programming Languages

Department: Computer Engineering

Instructor Information

Name: Assoc. Prof. Dr. Zeki Bayram

E-mail: zeki.bayram@emu.edu.tr

Office: CMPE 216

Office Tel: 0392 6302840

Assistant Information

Name:

E-mail:

Office:

Office Tel:

Meeting times and places

Tuesday 14:30-16:20 Room CMPE 128

Thursday 08:30-10:20 Room CMPE 128

Friday 08:30-10:30 Room CMPE 137 (Lab)

Program Name:

Computer Engineering / Software Engineering

Program Code:

25/29

Course Code

CMSE 318 – CMPE 410

Credits

4

Year/Semester

2019-2020 Spring

Required Course Elective Course (click on and check the appropriate box)

Prerequisite(s):

CMSE 211 – CMPE 211 Object-Oriented Programming

Catalog Description

Formal specification of programming languages: syntax, analysis, and semantics; evolution of programming languages and concepts; names and scope; data representation; evaluation sequence at expression, statement, and subprogram levels; Object Orientation implementation issues; abstraction, inheritance, polymorphism, concurrency, and exception handling; sampling of other paradigms such as functional, logical, scripting, high-performance, etc. as time permits. Weekly homework and lab work are assigned in parallel to lectures.

Course Web Page

<https://staff.emu.edu.tr/zekibayram/en/teaching/cmpe318>

Textbook(s)

SEBESTA, Robert W.: Concepts of Programming Languages, 10th Edition, Pearson Intl (Addison-Wesley), 2013. ISBN: 0-321-50968-4.

Indicative Basic Reading List

None.

**Topics Covered and Class Schedule
(4 hours of lectures per week)**

- Week 1** Introduction
Week 2 History
Week 3 Describing Syntax and Semantics
Week 4 Lexical and Syntax Analysis
Week 5 Names, Bindings, Type Checking, Scopes, Data Types
Week 6 Expressions and Assignment Statements
Week 7 Control Structures
Week 8 Functional Programming
Week 9 Subprograms
Week 10 Implementing Subprograms
Week 11 Abstract Data Types and Encapsulation Concepts
Week 12 Support for Object-Oriented Programming
Week 13 Concurrency (Time permitting)
Week 14 Exception Mechanism (Time permitting)

Lab Schedule

- Weeks 3-4** C++ data structures
Weeks 5-6 Lexical analysis
Weeks 7-8 Syntax analysis
Weeks 9-12 Haskell programming

Course Learning Outcomes

Upon successful completion of the course, students are expected to have the following competencies:

1. Draw an annotated parse tree for a given input and attribute grammar (SO 1)
2. Have knowledge of various programming languages, their features, history and category (SO 1)
3. Use LR parsing tables for bottom up parsing of a given input (SO 1)
4. Work effectively with context free grammars (SO 1)
5. Draw a parse tree for a sentence in a language, given its grammar (SO 1)
6. Derive a sentence in a language, given its grammar (SO 1)
7. Demonstrate that a specific grammar is ambiguous (SO 1)
8. Write a simple lexical analyzer (SO 1)
9. Write a simple top-down parser (SO 1)
10. Show the contents of the system stack after several function calls (SO 1)
11. Differentiate between static and dynamic scope (SO 1)
12. Trace output of programs with various parameter passing methods (SO 1)
13. Be familiar with the implementation techniques of object-oriented constructs (SO 1)
14. Write and trace simple programs in the Haskell Functional Programming Language (SO 1)

	Method	No	Percentage
Assessment	Midterm Exam(s)	1	--
	Final Examination	1	50%
	Attendance	-	5%
	Assignments	5	45%

Computation of the attendance grade: Proportional to the number of days in which the student attended lectures. Attendance will start to be taken once the add-drop period has ended.

Policy on makeups: For eligibility to take a makeup exam, the student should bring a doctor's report within 3 working days of the missed exam.

Policy on the NG grade: If you miss two exams with no valid excuse, you will be given the NG grade.

Policy on missed labs: There will be no makeup for missed labs. If you cannot attend a lab for some reason, you should contact the assistant *beforehand* so that you can present your work in advance.

Relationship of the course to ABET Student Outcomes

The course has been designed to contribute to the following student outcomes:

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

Prepared by: Assoc. Prof. Dr. Zeki Bayram

Date Updated: 20 April 2020