

**Arab Academy for Science and Technology and Maritime Transport
Computer Science Curriculum
Course Syllabus**

Course Code: CS445	Course Title: Structure of Programming Languages	Classification: R	Coordinator's Name: Prof. Dr. Mohamed Kholeif Lecturer's Name: Prof. Dr. Mohamed Kholeif	Credit Hours: 3
Pre-requisites: <ul style="list-style-type: none"> • CS311 (Theory of Computation) • CS321 (Systems Programming) 	Co-requisites: None	Schedule: Lecture: 2 hours Tutorial-Lab: 2 hours		
Office Hours: (Room 209) Tuesday 10:30 a.m. -12:30 p.m.				
Course Description: A concise introduction to the essentials of imperative programming languages, focusing on principles rather than specifics, while giving examples from many programming languages. The course also covers fundamental issues in language design. Overview of programming paradigms and type systems. Models of execution control: Order of evaluation of sub-expressions; conditional execution; iteration; exceptions and exception handling. Basic concepts of functional programming are also emphasized.				
Textbook: R. Sebesta, <i>Concepts of Programming Languages</i> , Addison- Wesley.				
References: H. Bal, D. Grune, <i>Programming Language Essentials</i> , Addison-Wesley.				

Course Objectives:	Contribution to Program Student Outcomes:
1. Be familiar with several language paradigms	SO1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Understand how languages relate to different application domains. 3. Understand the design space of programming languages, including concepts and constructs from past languages as well as those that may be used in the future. 4. Be able to develop a critical understanding of the programming language we use by being able to identify and compare the same concept as it appears in different languages.	SO1: Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. SO6: Apply computer science theory and software development fundamentals to produce computing-based solutions.
Course Outline:	
1. Week 1: Introduction to the course 2. Week 2: Preliminaries 3. Week 3: Describing Syntax and Semantics 4. Week 4: Describing Syntax and Semantics 5. Week 5: Lexical and Syntax Analysis 6. Week 6: Names, Bindings, Type Checking, and Scopes 7. Week 7: 7 th Week Examination 8. Week 8: Data Types	9. Week 9: Expressions and Assignment Statements 10. Week 10: Statement-Level Control Structures 11. Week 11: Statement-Level Control Structures 12. Week 12: 12 th Week Exam 13. Week 13: Subprograms 14. Week 14: Projects Submission 15. Week 15: Revision 16. Week 16: Final Examination
Grade Distribution: 7th Week Assessment (30%): Exam (20%) + Homework Assignments and/or quizzes 10% 12th Week Assessment (20%): Exam (15%) + Assignments 5% Year Work (10%): Course Project (10%) Final Exam (40%)	

Policies:

Attendance:

AASTMT Education and Study Regulations (available at aast.edu)

Academic Honesty:

AASTMT Education and Study Regulations (available at aast.edu)

Late Submission:

Late submissions are graded out of 75% (1 week late), 50% (2 weeks late), 25% (3 weeks late), 0% (more than 3 weeks late)