ISTANBUL MEDIPOL UNIVERSITY
SYLLABUS
IMU-COE4167930 PRINCIPLES OF PROGRAMMING LANGUAGES
2020 Fall Semester

Course Code	Course Name	Course Type	Weekly T	Weekly A	Weekly L	Credits	ECTS	Weekly Class Schedule
COE3149650	Principles of Programming Languages	Required	3	0	0	3	6	
Prerequisite	Object Oriented Programming	Prerequisite to						Thursday : 13:30-14:30 C-211 Online

Lecturer	Selim Akyokuş	Office Hours Schedule	Tuesday 14:30
E-mail	sakyokus@medipol.edu.tr		
Phone	5350	Office / Room No	C - 320 - North Campus
Assistants			
E-mail			

Course Objectives	The objective of this course is to study the properties of programming languages in general, learn syntax and semantics of programming languages, learn basic constructs that are common to all languages, examine some of these constructs and concepts for specific languages, introduce the main paradigms of computation, languages representative of these paradigms, techniques of implementing various programming language constructs, as well as basic concepts relating to the specification of programming languages.

Textbook	Required Textbooks: - Sebesta, Robert W. Concepts of Programming Languages, 11th ed, Addison-Wesley, 2017. Additional Textbooks and References: - Michael Scott. Programming Language Prgramatics, 4th edition, Morgan Kaufmann, San Francisco, California, 2015. - R. Toal, R. Rivera, A. Schneider, and E. Choe, Programming Language Explorations, CRC Press, 2017. - Pratt, T.W. & M.V.Zelkowitz. Programming Languages, Design and Implementation. Prentice Hall, 4th ed., 2001.

After successful completion of the course, the student will be able to:

	Learning Outcomes
1	Discuss about evolution of programming languages and the role of procedural, object-oriented, functional, declarative, scripting languages
2	Understand syntax, semantic, lexical and syntax analysis
3	Describe in detail the design issues for the primary constructs of the imperative languages
4	Describe control statements, discuss subprograms and their implementations
5	Describe the principles and constructs of object-oriented languages.
6	Have the ability to learn and choose new languages for an area of applicaton easily
7	Have an understanding of features that should be included in a new language they design

Teaching Methods	Class discussions with examples. The notes and the presentations will be delivered during the lectures.

WEEK	TOPIC	REFERENCE
Week 1	Introduction	Slides and Sebesta Chap. 1
Week 2	Evolution of Major Programming Languages	Slides and Sebesta Chap. 2
Week 3	Describing Syntax and Semantics	Slides and Sebesta Chap. 3
Week 4	Lexical and Syntax Analysis	Slides and Sebesta Chap. 4
Week 5	Names, Bindings, Type Checking, and Scopes	Slides and Sebesta Chap. 5
Week 6	Data Types	Slides and Sebesta Chap. 6
Week 7	Expressions and Assignment Statements	Slides and Sebesta Chap. 7
Week 8	Exam Week	All Slides and Chapters till Week 8
Week 9	Statement-Level Control Structures and Subprograms	Slides and Sebesta Chap. 8, 9
Week 10	Implementing Subprograms, Abstract Data Types and Encapsulation Constructs	Slides and Sebesta Chap. 10,11
Week 11	Support for Object-Oriented Programming	Slides and Sebesta Chap. 12
Week 12	Concurrency	Slides and Sebesta Chap. 13
Week 13	Exception and Event Handling	Slides and Sebesta Chap. 14
Week 14	Functional and Logic Programming Languages	Slides and Sebesta Chap. 15,16

Assessment Methods and Criteria	Evaluation Tool	Quantity	Weight
	Final Exam	1	45%
	Midterm	-	35%
	Quizes	-	
	HW Assignments & Projects	-	20%

***   ECTS   Language of Instruction: English

Activity	Hours	Weeks	Student Workload Hours	Activity	Hours	Weeks	Student Workload Hours
Lecture hours	3	14	42.0	In-term exam study	20	2	40.0
HWs	5	4	20.0	Final exam study	30	1	30.0
Projects	8	3	24.0				
						Total Workload Hours =	156.0
						Recommended ECTS Credit =	6