

<b>Course Description</b>					
<b>Name</b>	<b>Code</b>	<b>Semester</b>	<b>T+A Hour</b>	<b>Credit</b>	<b>ECTS</b>
PRINCIPLES of PROGRAMMING LANGUAGES	COE3167930	Fall Semester	3+0	3	6
<b>Prerequisites Courses</b>	NESNE TABANLI PROGRAMLAMA				
<b>Recommended Elective Courses</b>					
<b>Language of Instruction</b>	English				
<b>Course Level</b>	First Cycle (Bachelor's Degree)				
<b>Course Type</b>	Required				
<b>Course Coordinator</b>	Prof.Dr. Selim AKYOKUŞ				
<b>Name of Lecturer(s)</b>	Prof.Dr. Selim AKYOKUŞ				
<b>Assistant(s)</b>					
<b>Aim</b>	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.				
<b>Course Content</b>	This course contains; Introduction, Evolution of Major Programming Languages, Describing Syntax and Semantics, Lexical and Syntax Analysis, Names, Bindings, Type Checking, and Scopes, Data Types, Expressions and Assignment Statements, Exam Week, İfade-Seviyesi Kontrol Yapıları ve Altprogramlar , Implementing Subprograms, Abstract Data Types and Encapsulation Constructs, Support for Object-Oriented Programming, Concurrency, Exception and Event Handling, Functional and Logic Programming Languages.				
<b>Course Learning Outcomes</b>			<b>Teaching Methods</b>	<b>Assessment Methods</b>	
Understand syntax, semantic, lexical and syntax analysis					
Understand and Discuss about evolution of programming languages and the role of procedural, object-oriented, functional, declarative, scripting languages.					
Describe in detail the design issues for the primary constructs of the imperative languages					
Understand and Describe control statements, discuss subprograms and their implementations, describe the principles and constructs of object-oriented languages.					
Have the ability to learn and choose new languages for an area of application easily.					
Have ideas about features that should be included in the design of a new language.					
<b>Teaching Methods</b>					
<b>Assessment Methods</b>					
<b>Lecture Schedule</b>					
<b>Sequence</b>	<b>Topics</b>	<b>Preliminary Preparation</b>			
1	Introduction	Slides and Sebesta Chap. 1			
2	Evolution of Major Programming Languages	Slides and Sebesta Chap. 2			
3	Describing Syntax and Semantics	Slides and Sebesta Chap. 3			
4	Lexical and Syntax Analysis	Slides and Sebesta Chap. 4			
5	Names, Bindings, Type Checking, and Scopes	Slides and Sebesta Chap. 5			
6	Data Types	Slides and Sebesta Chap. 6			
7	Expressions and Assignment Statements	Slides and Sebesta Chap. 7			
8	Exam Week	All Slides and Chapters till Week 8			
9	İfade-Seviyesi Kontrol Yapıları ve Altprogramlar	Slides and Sebesta Chap. 8, 9			
10	Implementing Subprograms, Abstract Data Types and Encapsulation Constructs	Slides and Sebesta Chap. 10,11			
11	Support for Object-Oriented Programming	Slides and Sebesta Chap. 12			
12	Concurrency	Slides and Sebesta Chap. 13			
13	Exception and Event Handling	Slides and Sebesta Chap. 14			
14	Functional and Logic Programming Languages	Slides and Sebesta Chap. 15,16			
<b>Evaluation Methods</b>		<b>Weight(%)</b>			
Midterm Exam		30			
General Exam		70			

<b>Resources</b>
- Sebesta, Robert W. Concepts of Programming Languages, 11th ed, Addison-Wesley, 2017.
- Michael Scott. Programming Language Pragmatics, 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.