Course Description Form	
Course Code and Name	BM224 PROGRAMMING LANGUAGES
Course Semester	4
Catalog Content	Conceptual study on syntax, semantics and application of programming languages, Lambda analysis and functional languages, basic notions of expressive semantics and language features, the theory of autocorrelation, first order logic and declarative languages, harmonization of semantic definitions.
Textbook	Sebesta, R. W., & Mukherjee, S. (2015). Concepts of programming languages (Vol. 8). Addison-Wesley.
Supplementary Textbooks	Darnell, P. A., & Margolis, P. E. (2012). Software engineering in C. Springer Science & Business Media. Comparative Programming Languages (3rd Edition) by Robert G. Clark, 2000.
Credit	6
Prerequisites of the Course (*Attendance Requirements*)	Prerequisites course: No Co-requisites: Obligatory course attendance 70%
Type of the Course	Compulsory
Instruction Language	Turkish
Course Objectives	Teaching the design of programming languages and compare alternative designs by approaching from different angles. To increase the familiarity to different programming languages. by examining the various examples of different programming languages.
Course Learning Outcomes	Defining of programming languages syntax, their meanings and usage Formal syntactic representation, expression structure, postfix and infix representation and conversion Data types and variables, expression and assignment statements, control structures and subroutines Functional and logical languages and programming Current programming languages key properties Program structures, procedures, functions and methods Inheritance and dynamic delimiters Matching, consistency and network structure Graphical user interfaces
Instruction Methods	The mode of delivery of this course is face to face

Weekly Schedule	1. week: Defining of programming languages syntax, their meanings and usage 2. week: Formal syntactic representation, expression structure, postfix and infix representation and conversion 3. week: Data types and variables, expression and assignment statements, control structures and subroutines 4. week: Functional and logical languages and programming 5. week: Current programming languages key properties 6. week: Program structures 7. week: Procedures 8. week: Procedures 9. week: Functions and methods 10. week: Inheritance and dynamic delimiters 11. week: Inheritance and dynamic delimiters 12. week: Matching, consistency and network structure 13. week: Graphical user interfaces 14. week: Graphical user interfaces
Teaching and Learning Methods *(These are examples. Please fill which activities you use in the course)*	Weekly theoretical course hours: 3 Reading Searching in Internet and library Preparing presentation Presentation Midterm and studying for midterm Final and studying for final

Assessment Criteria

	Numbers	Total Weighting (%)	
Midterm Exams	1	30	
Assignment	0	0	
Application	0	0	
Projects	1	30	
Practice	0	0	
Quiz	0	0	
Percent of In-term Studies (%)		60	
Percentage of Final Exam to Total Score (%)		40	
Attendance		-	

	Activity	Total Number of Weeks	Duration (weekly hour)	Total Period Work Load
Workload	Weekly Theoretical Course Hours	14	3	42
	Weekly Tutorial Hours	0	0	0
	Reading Tasks	10	4	40
	Studies	10	3	30
	Material Design and Implementation	0	0	0
	Report Preparing	0	0	0
	Preparing a Presentation	1	12	12
	Presentations	1	1	1
	Midterm Exam and Preparation for Midterm Exam	1	10	10
	Final Exam and Preparation for Final Exam	1	15	15
	Other (should be emphasized)			0
	Total Workload			150
	Total Workload / 25			6
	Course Credit (ECTS)			6

	No	Program Outcomes	1	2	3	4	5
Contribution Level Between Course Learning Outcomes and Program Outcomes	1	Sufficient knowledge on mathematics, science and computer engineering; ability to apply theoretical and practical knowledge in these areas to model and solve engineering problems		X			
	2	Ability to identify, define, formulate and solve complex engineering problems; ability to choose and apply appropriate analysis and modelling methods for these purposes			X		
	3	Ability to design a complex system, process, device, software, algorithm, or product under realistic constraints and circumstances to meet certain requirements; ability to apply modern design techniques for this purpose				X	
	4	Ability to choose, develop and use modern techniques and tools necessary for engineering applications; ability to effectively use computing technologies					X
	5	Ability to design and implement systems or experiments to solve engineering problems, collect and interpret data to evaluate and analyze the results of solutions			X		
	6	Ability to work effectively in intradisciplinary and interdisciplinary teams or individually		X			

	7	Ability to efficiently prepare, evaluate and interpret reports		X		
	8	Ability to make presentations and conduct effective verbal and written communication in Turkish and English		X		
	9	Awareness of the necessity of lifelong learning; ability to access information, follow scientific and technological developments; ability to perpetually renew oneself				X
	10	Awareness of professional and ethical responsibility, ability to act in accordance with ethical principles		X		
	11	Ability to apply knowledge on project management, risk management and change management			X	
	12	Awareness of entrepreneurship and innovation, ability to design and build sustainable systems		X		
	13	Ability to devise local and global solutions to contemporary issues considering the effects of engineering applications on health, environment and security			X	
	14	Awareness of the legal consequences of engineering solutions	X			
	15	Ability to apply knowledge on software development process and documentation rules			X	
	16	Knowledge on standards used in engineering applications				X
	17	Awareness of occupational health and security, information security and privacy	X			
The Course's Lecturer(s) and Contact Information	Computer Engineering Department Chair bmbb@gazi.edu.tr					