



MARMARA UNIVERSITY - FACULTY OF ENGINEERING

2017-2018 Fall

CSE 3063 Object Oriented Software Design

COURSE DESCRIPTION FORM

Offering Department		Department of Computer Engineering		Undergraduate must course (5th semester)							
Course Code		CSE3063									
Course Name		Object Oriented Software Design									
Language of Instruction		English									
ECTS		5									
Contact Hours		Theoretical (T): 3			Practice (U):			Laboratory(L):			
Pre-requisites		CSE1142 Computer Programming II									
Instructor		Name		Murat Can Ganiz							
		E-mail		murat.ganiz@marmara.edu.tr							
Course Materials		Mandatory		Craig Larman, Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design, Prentice Hall, 2005. UML Distilled (Third Edition) by Martin Fowler ISBN 0-321-19368-7							
		Recommended		Object-Oriented Analysis and Design with Applications (Third Edition) by Grady Booch ISBN 0-201-89551-X Design Patterns (Elements of Reusable Object-Oriented Software) by Erich Gamma ISBN 0-201-63361-2							
Course Objectives		To provide students with a broad perspective of what Object Oriented Design (OOP) means, how it relates to UML, how it differs from the procedural programming paradigm, to convince students that software design for successful software projects is important, and to demonstrate the importance of successful communication in software development, and to show that UML is a powerful tool for communication. To enable students to apply Object Oriented Design principles and software patterns in practical projects.									
Course Content		Basic concepts of object oriented programming, general features of Object Oriented Programming languages, software development processes and basic information about the advantages of iterative processes such as Scrum, information about software development stages; analysis, design, coding, and testing activities and their place in software development processes, using UML for object oriented analysis and design, UML class and sequence diagram types, analysis process and tools used in this process, design process and tools used in this process, introduction to Python programming language, object oriented programming in Python programming language, class writing and inheritance operations, responsibility-driven object design principle, information about GRASP software design patterns for responsibility assignments and for expandable / reusable design, introduction of GoF software design patterns									
Learning Outcomes		LO1		To be able to explain the distinction between object oriented and procedural programming approaches and the advantages of object oriented approach.							
		LO2		To be able to write a software requirement analysis document individually or in a team by incorporating domain class diagrams and system sequence diagrams							
		LO3		To be able to create UML class and sequence diagrams individually or in a team.							
		LO4		To be able to understand software design tools and software design tools such as UML Class and Sequence diagrams and convert them into code effectively individually or in a team.							
		LO5		To learn the basic features of the Python programming language and gain the ability to create projects in this language.							
Program Outcomes				LO1	LO2	LO3	LO4	LO5	LO6	LO7	
PO2		Ability to identify, formulate, and solve complex engineering problems (a); ability to select and apply proper analysis and modelling methods for this purpose (b).			a	b	b				
PO4		Ability to devise (a), select, and use (b) modern techniques and tools needed for engineering practice (1); ability to employ information technologies effectively (2).		2	1a,1b		1a,1b	2			
PO6		Ability to work efficiently in intra-disciplinary (a) and multi-disciplinary teams (b); ability to work individually (c).			a,c	a,c	a,c	a,c			
PO10		Information about business life practices such as project management, risk management, and change management (a); awareness of entrepreneurship, innovation (b), and sustainable development (c).					a				
PO13		Knowledge of mathematics, basic sciences (a), computer science (b) and engineering sciences (c) required for the design and analysis of complex electrical and electronic devices, software and systems including hardware and software.			b	b	b				
Subjects (Knowledge,		No	Week	Subjects	LO1	LO2	LO3	LO4	LO5	LO6	LO7
S1		1-2	Software Project Management		HMF						

Skills and Behaviours), Contributions of Subjects to Learning Outcomes, Assessment Methods	S2	3	Object Oriented Programming approach	HMF					
	S3	4	Software requirements analysis.		M				
	S4	5	Domain class and system sequence diagrams for software requirement analysis		M				
	S5	6	UML class and object diagrams for software design			FP	P		
	S6	7-9	UML sequence diagrams for software design			FP	P		
	S7	10	Python Programming language general features and differences from C ++ / Java programming languages					P	
	S8	11	Object oriented programming in Python Programming language, classes, encapsulation, inheritance					P	
	S9	12-14	Software Design Patterns					P	

Assessment Methods and Weights	No	Type	Weight	Implementation Rule	Make-up Rule
	MF	Midterm, Final	60%	Exams, are closed book. No calculation and communication tools are allowed during exams.	Marmara University regulations will be followed for make-up exams.
	Q	Quiz	-	-	-
	H	Homeworks	10%	The deadline for homework is one week before the oral exams. Oral exams are held randomly from the attendance list every week.	-
	P	Project	30%	The first project is done in Java with 3 UP / Scrum iterations (20%). The second project is done in Python with a single iteration (10%).	-
	TOTAL		100%		

Determining Letter Grades	<ul style="list-style-type: none"> The letter grades will be determined based on the midterm and final exams, projects and homeworks. In order to determine the letter grade, a curve or catalog based method will be followed based on the total average scores of the students. The final exam score and the total average score of the student must be at least 35 to pass the course. According to Marmara University Undergraduate regulations, the weight of the final exam must be at least 40 out of 100. 											
	<table border="1"> <thead> <tr> <th>Assessment</th> <th>Midterm</th> <th>Project</th> <th>Homeworks</th> <th>Final</th> <th>TOTAL</th> </tr> </thead> <tbody> <tr> <td>Weight</td> <td>20</td> <td>30</td> <td>10</td> <td>40</td> <td>100</td> </tr> </tbody> </table>	Assessment	Midterm	Project	Homeworks	Final	TOTAL	Weight	20	30	10	40
Assessment	Midterm	Project	Homeworks	Final	TOTAL							
Weight	20	30	10	40	100							

Teaching Method, Student Work Load	Time Applied by the Instructor			
	No	Method	Explanation	Hours
	1	Lectures	Lectures are given in class using the board or via presentations. Example questions are solved to enhance the concepts.	14x3=42
	2	Problem Session/ Practice	Problems related to the course topics are solved on the board.	
	3	Laboratory	Experiments are done in the laboratory or theoretical concepts covered during the lectures are practiced using computer exercises.	
	4	Interactive Courses	Questions are asked to students during lectures and they are encouraged to guess the answers (peer learning is also in this category)	
	5	Field Work	Students attend activities outside the campus.	
	6	Midterm	Midterm exam is given during the midterm week.	2
	7	Final	Final exam is given during the final exam week.	2
	Estimated Time to be Allocated by a Student			
	8	Project	The students carry out research about the problem given in the project, design and implement their solution and prepare a report.	2x20=40
	9	Homeworks	The students solve the problems given as homework.	5
	10	Pre-class learning of Course Material	The students study and learn the new subjects from course materials.	
11	Review of Course Material	Students review the course subjects from course materials to prepare for the exams and homeworks.	36	
12	Office Hour	Students ask questions to the instructor or the assistant during office hours.	2	
TOTAL			129	

Academic Honesty	Violations of scholastic honesty include, but are not limited to cheating, plagiarizing, fabricating information or citations, facilitating acts of dishonesty by others, having unauthorized possession of examinations, submitting work of another person or work previously used without informing the instructor, or tampering with the academic work of other students.
	In case academic dishonesty is observed, the first authority is the instructor of the course. The instructor may decide to give the student zero for the homework(s)/lab(s)/exam(s), give the letter grade FF, or may take disciplinary action.