

**İTÜ**  
**LİSANSÜSTÜ DERS KATALOG FORMU**  
**(GRADUATE COURSE CATALOGUE FORM)**

<b>Dersin Adı</b>		<b>Course Name</b>		
YAZILIM MODELLEME VE TASARIMI		SOFTWARE MODELING AND DESIGN		
<b>Kodu (Code)</b>	<b>Yarıyıl (Semester)</b>	<b>Kredisi (Local Credits)</b>	<b>AKTS Kredisi (ECTS Credits)</b>	<b>Ders Türü (Course Type)</b>
BBL 502	Güz	3	7.5	Zorunlu (Compulsory)
<b>Bölüm / Program (Department/Program)</b>	Bilgisayar Bilimleri (Computer Science)			
<b>Dersin Türü (Course Type)</b>	Zorunlu, Meslek, Compulsory, Engineering	<b>Dersin Dili (Course Language)</b>	Türkçe (Turkish)	
<b>Dersin İçeriği (Course Description)</b>	İsteklerin ve problemin çözümlenmesi. Sistemin, işbirliği yapan nesnelere dayalı olarak nasıl tasarlanması Analiz ve tasarımların ifade UML kullanılması. GRASP kalıpları ve yaygın biçimde kabul gören GoF tasarım kalıpları Yazılım kalitesinin ölçülmesinde ve değerlendirilmesinde kullanılan yazılım metrikleri. <i>30-60 kelime arası</i>			
	Requirement and problem domain analysis. Object oriented design Using the UML to express software artifacts. GRASP and GoF software design patterns Metrics to evaluate the design quality of software			
<b>Dersin Amacı (Course Objectives)</b>	1. İstekleri anlamak için senaryo yazılabilmesi. 2. Problem uzayının analiz edilebilmesi 3. Bir yazılım sisteminin nesneye dayalı olarak tasarlanabilmesi 4. Önemli yazılım tasarım kalıplarının öğretilmesi			
	1. Writing use cases to model requirements 2. Analysis of the problem domain 3. Designing a software system using the object oriented approach 4. Teaching some of the important software design patterns			
<b>Dersin Öğrenme Çıktıları (Course Learning Outcomes)</b>	1. Öğrenciler yazılım kullanım durumları yazabilirler. 2. Öğrenciler problem uzayını analiz edilebilirler 3. Öğrenciler bir yazılım sisteminin nesneye dayalı olarak tasarlanmasında neler yapılması gerektiğini bilirler. 4. Öğrenciler önemli yazılım tasarım kalıplarını tanırlar ve kullanılmasını bilirler 5. Öğrenciler önemli yazılım değerlendirme metriklerini bilirler.			
	1. Students can write software use cases. 2. Students can analyze the problem domain of a software system. 3. Students know the road map how to design an object oriented software system. 4. Students are familiar with the important software design patterns and can use them. 5. Students are familiar with the important software evaluation metrics.			

<b>Ders Kitabı (Textbook)</b>	Craig Larman, Applying UML and Patterns , An Introduction to OOA/D and Iterative Development, 3/e, Prentice Hall PTR, 2005.
<b>Diğer Kaynaklar (Other References)</b>	
	<i>Maddeler halinde en çok 5 adet</i>
<b>Ödevler ve Projeler (Homework &amp; Projects)</b>	<b>TÜM YAZILIM GELİŞTİRME AŞAMALARINI İÇEREN 1 ADET DÖNEM PROJESİ</b> <b>A SOFTWARE DESIGN PROJECT THAT INCLUDES ALL DESIGN PHASES</b>

<b>Laboratuvar Uygulamaları</b> (Laboratory Work)			
<b>Bilgisayar Kullanımı</b> (Computer Use)			
<b>Diğer Uygulamalar</b> (Other Activities)			
<b>Başarı Değerlendirme Sistemi</b> (Assessment Criteria)	<b>Faaliyetler</b> (Activities)	<b>Adedi*</b> (Quantity)	<b>Değerlendirmedeki Katkısı, %</b> (Effects on Grading, %)
	<b>Yıl İçi Sınavları</b> (Midterm Exams)	<b>1</b>	<b>% 350</b> (35 %)
	<b>Kısa Sınavlar</b> (Quizzes)		
	<b>Ödevler</b> (Homework)		
	<b>Projeler</b> (Projects)	<b>1</b>	<b>% 25</b> (25 %)
	<b>Dönem Ödevi/Projesi</b> (Term Paper/Project)		
	<b>Laboratuvar Uygulaması</b> (Laboratory Work)		
	<b>Diğer Uygulamalar</b> (Other Activities)		
	<b>Final Sınavı</b> (Final Exam)	<b>1</b>	<b>% 40</b> (40 %)

\*Yukarıda Belirtilen Sayılar Minimum Olup Yerine Getirilmesi Zorunludur.

## DERS PLANI

Hafta	Konular	Dersin Çıktıları
1	GİRİŞ Nesneye Dayalı Programlama temel kavramlar	1,2,3
2	Tümleştirilmiş Yazılım Geliştirme Süreci	1,2,3
3	Kullanım Durumları	1
4	Problem Uzayının Modellenmesi, Analiz	2
5	UML sınıf ve etkileşim diyagramları	2,3
6	Yazılım tasarımın aşamaları	3
7	GRASP Kalıpları (4 adet)	4
8	GRASP Kalıpları kullanılarak örnek tasarımları yapılması	3,4
9	Kodlama ve birim sınama	3,4
10	GRASP Kalıpları (devamı, 5 adet)	4
11	GoF Tasarım Kalıpları (5 adet)	4
12	GoF Tasarım Kalıpları (devamı, 5 adet)	4
13	GoF Tasarım Kalıpları kullanılarak örnek tasarımları yapılması	3,4
14	Yazılım değerlendirilmede kullanılan temel ölçütler	5

## COURSE PLAN

Weeks	Topics	Course Outcomes
1	Introduction, Basic information on OOP	1,2,3
2	The Unified Process	1,2,3
3	Use Cases	1
4	Modeling of the problem domain, analysis	2
5	UML class and interaction diagrams	2,3
6	Phases of the software design	3
7	GRASP design patterns (4 patterns)	4
8	Example designs using GRASP patterns	3,4
9	Coding and unit tests	3,4
10	GRASP design patterns (cont. 5 patterns)	3
11	GoF design patterns (5 patterns)	3
12	GoF design patterns (cont. 5 patterns)	3
13	Example designs using GoF patterns	3,4
14	Basic metrics for software evaluation	5

## Dersin Bilgisayar Mühendisliği Yüksek Lisans Programıyla İlişkisi

	Programın mezuna kazandıracığı bilgi, beceri ve yetkinlikler (programa ait çıktılar)	Katkı Seviyesi		
		1	2	3
i.	Lisans düzeyi yeterliliklerine dayalı olarak, Bilgisayar Mühendisliği alanında bilgilerini uzmanlık düzeyinde geliştirebilme ve derinleştirebilme (yeterli bilgi birikimi) (bilgi).			X
ii.	Bilgisayar Mühendisliği alanının ilişkili olduğu disiplinler arası etkileşimi kavrayabilme (bilgi).			
iii.	Bilgisayar Mühendisliği alanında edindiği uzmanlık düzeyindeki kuramsal ve uygulamalı bilgileri kullanabilme (beceri).		X	
iv.	Bilgisayar Mühendisliği alanında edindiği bilgileri farklı disiplin alanlarından gelen bilgilerle bütünleştirerek yorumlayabilme ve yeni bilgiler oluşturabilme (beceri).			
v.	Bilgisayar Mühendisliği alanını ile ilgili karşılaşılan sorunları araştırma yöntemlerini kullanarak çözümlenebilme (beceri).			
vi.	Bilgisayar Mühendisliği alanı ile ilgili uzmanlık gerektiren bir çalışmayı bağımsız olarak yürütebilme (Bağımsız Çalışabilme ve Sorumluluk Alabilme Yetkinliği).		X	
vii.	Bilgisayar Mühendisliği alanı ile ilgili uygulamalarda karşılaşılan ve öngörülemeyen karmaşık sorunların çözümü için yeni stratejik yaklaşımlar geliştirebilme ve sorumluluk alarak çözüm üretebilme (Bağımsız Çalışabilme ve Sorumluluk Alabilme Yetkinliği).			
viii.	Bilgisayar Mühendisliği alanı ile ilgili sorunların çözümlenmesini gerektiren ortamlarda liderlik yapabilme (Bağımsız Çalışabilme ve Sorumluluk Alabilme Yetkinliği).			
ix.	Bilgisayar Mühendisliği alanında edindiği uzmanlık düzeyindeki bilgi ve becerileri eleştirel bir yaklaşımla değerlendirebilme ve öğrenmesini yönlendirebilme (Öğrenme Yetkinliği).			
x.	Bilgisayar Mühendisliği alanındaki güncel gelişmeleri ve kendi çalışmalarını, nicel ve nitel veriler ile destekleyerek, alanındaki ve alan dışındaki gruplara, yazılı, sözlü ve görsel olarak sistemli biçimde Türkçe ve/veya İngilizce olarak aktarabilme (İletişim ve Sosyal Yetkinlik).			
xi.	Sosyal ilişkileri ve bu ilişkileri yönlendiren normları eleştirel bir bakış açısı ile inceleyebilme, geliştirebilme ve gerektiğinde değiştirmek üzere harekete geçebilme (İletişim ve Sosyal Yetkinlik).			
xii.	Bilgisayar Mühendisliği alanının gerektirdiği düzeyde bilgisayar yazılımı ile birlikte bilişim ve iletişim teknolojilerini ileri düzeyde kullanabilme (İletişim ve Sosyal Yetkinlik).			
xiii.	Bilgisayar Mühendisliği alanı ile ilgili verilerin toplanması, yorumlanması, uygulanması ve duyurulması aşamalarında toplumsal, bilimsel, kültürel ve etik değerleri gözetenerek denetleyebilme ve bu değerleri öğretebilme (Alana Özgü Yetkinlik).			
xiv.	Bilgisayar Mühendisliği alanı ile ilgili konularda strateji, politika ve uygulama planları geliştirebilme ve elde edilen sonuçları, kalite süreçleri çerçevesinde değerlendirebilme (Alana Özgü Yetkinlik).			
xv.	Bilgisayar Mühendisliği alanında özümledikleri bilgiyi, problem çözme ve/veya uygulama becerilerini, disiplinlerarası çalışmalarda kullanabilme (Alana Özgü Yetkinlik).			
xvi.	Tezli programlarda, kendi çalışmalarını, Bilgisayar Mühendisliği alanındaki uluslararası platformlarda, yazılı, sözlü ve/veya görsel olarak aktarabilme (Alana özgü yetkinlik).			

1: Az, 2. Kısmi, 3. Tam

## Relationship between the Course and Computer Engineering Graduate (MS) Curriculum

	Program Outcomes	Level of Contribution		
		1	2	3
i.	Developing and intensifying knowledge in Computer Engineering area, based upon the competency in the undergraduate level (sufficient knowledge) (knowledge).			X
ii.	Grasping the inter-disciplinary interaction related to Computer Engineering area (knowledge).			
iii.	The ability to use the expert-level theoretical and practical knowledge acquired in Computer Engineering area (skill).		X	
iv.	Interpreting and forming new types of knowledge by combining the knowledge from Computer Engineering area and the knowledge from various other disciplines (skill).			
v.	Solving the problems faced in Computer Engineering area by making use of the research methods (skill).			
vi.	The ability to carry out a specialistic study related to Computer Engineering area independently (Competence to work independently and take responsibility).		X	
vii.	Developing new strategic approaches to solve the unforeseen and complex problems arising in the practical processes of Computer Engineering area and coming up with solutions while taking responsibility (Competence to work independently and take responsibility).			
viii.	Fulfilling the leader role in the environments where solutions are sought for the problems related to Computer Engineering area (Competence to work independently and take responsibility)			
ix.	Assessing the specialistic knowledge and skill gained through the study with a critical view and directing one's own learning process (Learning Competence).			
x.	Systematically transferring the current developments in Computer Engineering area and one's own work to other groups in and out of Computer Engineering area; in written, oral and visual forms in turkish and/or english (Communication and Social Competency).			
xi.	Ability to see and develop social relationships and the norms directing these relationships with a critical look and the ability to take action to change these when necessary. (Communication and Social Competency).			
xii.	Using the computer software together with the information and communication technologies efficiently and according to the needs of Computer Engineering area (Communication and Social Competency).			
xiii.	Paying regard to social, scientific, cultural and ethical values while collecting, interpreting, practicing and announcing processes of Computer Engineering area related data and the ability to teach these values to others (Area Specific Competency).			
xiv.	Developing strategy, policy and application plans concerning the subjects related to Computer Engineering area and the ability to evaluate the end results of these plans within the frame of quality processes (Area Specific Competency).			
xv.	Using the knowledge and the skills for problem solving and/or application (which are processed within the area) in inter-disciplinary studies (Area Specific Competency).			
xvi.	In the programs with thesis, the ability to present one's own work within the international Computer Engineering environments orally, visually and in written forms (Area Specific Competency).			

**1: Little, 2. Partial, 3. Full**

<u><i>Düzenleyen (Prepared by)</i></u> Feza BUZLUCA	<u><i>Tarih (Date)</i></u> 13. 06. 2011	<u><i>İmza (Signature)</i></u>
--	--	--------------------------------