

<b>Dersin Adı</b>	<b>Kodu</b>	<b>Yarıyılı</b>	<b>T+U</b>	<b>Kredisi</b>	<b>AKTS</b>
Yazılım Tasarımı ve Mimarisi		3	3+0	3	6
<b>Ön koşul Dersler</b>	Yok				
<b>Dersin Dili</b>	Türkçe				
<b>Dersin Seviyesi</b>	Lisans				
<b>Dersin Türü</b>	Zorunlu				
<b>Dersin Koordinatörü</b>					
<b>Dersi Verenler</b>					
<b>Dersin Yardımcıları</b>					
<b>Dersin Amacı</b>	Bu dersin amacı, yazılım projelerinin yazılım mühendisliği disipliniyle modellenmesini, tasarlanmasını ve geliştirilmesini sağlamaktır. Bu amaçla yazılım mühendisliğinin temel prensiplerini, yazılım sürecinin adımlarını, yazılım tasarımının standartlarını ve standart haline gelmiş tasarım kalıplarını öğretmektir.				
<b>Dersin İçeriği</b>	Bu ders yazılım mimarisi ve tasarımı için temel tasarım ilkeleri ve stratejileri kapsar. Mimari stiller, kalite nitelikleri, gösterimler ve belgeler, referans mimarisi, mimari sürecinde etki özgü mimarisi ve desen odaklı tasarım, bileşen tabanlı tasarım, boy odaklı tasarım ve detay tasarım sürecinde arayüz tasarımı ele alınmıştır.				
<b>Dersin Öğrenme Kazanımları</b>	<p>Bu dersi başarıyla tamamlayan öğrenci;</p> <ol style="list-style-type: none"> <li>1. Yazılım Mühendisliğinin temel elemanlarını tanımlayabilir.</li> <li>2. Yazılım Tasarımı ve Mimari temellerini anlayabilir.</li> <li>3. Mimari tasarım yöntemlerini anlama, araştırma yapabilir ve karşılaştırabilir.</li> <li>4. Gerçek hayat projesi geliştirebilir.</li> </ol>				
<b>Haftalar</b>	<b>Konular</b>				
1	Yazılım Mimarisi Nedir				
2	Yazılım Mimarisi Neden Önemlidir?				
3	Yazılım mimarisinin kapsamı				
4	Kalite Gereksinimleri Bölüm I				
5	Kalite Gereksinimleri Bölüm II				
6	Çevik Projelerde Yazılım Mimarisi				
7	Mimari ve Gereksinimler				
8	Mimariyi tasarlamak				
9	Yazılım Mimarisi Dökümantasyonu				
10	Yazılım Mimarisi Vaka Çalışması I				
11	Yazılım Mimarisi Vaka Çalışması II				
12	Bulutta Mimari				
13	Mimaride yeni yaklaşımlar				
14	Belirtim ve tasarım araçları ve sistem testleri geliştirmek için alternatif tasarım yaklaşımları.				
15	Proje Sunumları				
<b>Genel Yeterlilikler</b>					
Verilen herhangi bir yazılım mühendisliği teknik kavramı konularında araştırma ve öğrenmeyi en doğru şekilde yapabilmek.					
<b>Kaynaklar</b>					
Software Architecture, Kazman et al.					
<b>Değerlendirme Sistemi</b>					
Ara Sınav, Kısa Sınav, Yarıyıl Sonu Sınavı ve Değerlendirmelerin yapılacağı tarih, gün ve saatler daha sonra Fakülte Yönetim Kurulunun alacağı karara göre açıklanacaktır.					

PROGRAM ÖĞRENME ÇIKTILARI İLE DERS ÖĞRENİM KAZANIMLARI İLİŞKİSİ TABLOSU											
	PÇ1	PÇ2	PÇ3	PÇ4	PÇ5	PÇ6	PÇ7	PÇ8	PÇ9	PÇ10	PÇ11
ÖK1	5	5	5	5	4	4	4	5	5	4	4
ÖK2	5	4	4	4	4	3	3	3	5	4	5
ÖK3	5	5	3	4	5	3	3	3	5	3	5
ÖK4	5	5	3	4	3	3	3	3	5	4	5
ÖK: Öğrenme Kazanımları PÇ: Program Çıktıları											
Katkı Düzeyi	1 Çok Düşük		2 Düşük		3 Orta		4 Yüksek		5 Çok Yüksek		

**Program Çıktıları ve İlgili Dersin İlişkisi**

	PÇ1	PÇ2	PÇ3	PÇ4	PÇ5	PÇ6	PÇ7	PÇ8	PÇ9	PÇ10	PÇ11
Yazılım Tasarımı ve Mimarisi	5	4	5	5	3	3	3	4	5	4	5