

GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	Advanced Software Architecture / Advanced Software Architecture	
Ders Kodu / Course Code	BLG515	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	Master with Thesis / Master with Thesis	
Ders Akts Kredi / ECTS	6.00	
Haftalık Ders Saati (Kuramsal) / Course Hours For Week (Theoretical)	0.00	
Haftalık Uygulama Saati / Course Hours For Week (Objected)	0.00	
Haftalık Laboratuvar Saati / Course Hours For Week (Laboratory)	0.00	
Dersin Verildiği Yıl / Year	1	
Öğretim Sistemi / Teaching System	Daytime Class / Daytime Class	
Eğitim Dili / Education Language		
Ön Koşulu Olan Ders(ler) / Precondition Courses	yok	None
Amacı / Purpose	Bu ders, yazılım mimarisinin temel kavramını, mimari stillerini, yazılım mimarisini modelleme yöntemlerini, yazılım mimarisi tanımlama dillerini ve XCD yazılım mimarisi tanımlama dilini kapsamaktadır. Ayrıca, yazılım mimarisinin görselleştirilmesi, UML yazılım tasarım dili, yazılım mimarisinin program koduna dönüştürülmesi, yazılım mimarisinin fonksiyonel olmayan sistem özelliklerine göre tanımlanması ve alan bazlı yazılım mimarisi geliştirme konuları da dersin içeriğini oluşturmaktadır. Bu dersin amacı, öğrencilere yazılım mimarisini kavramsal, ilkeleri ve teknikleri açısından tanıtmaktır. Aynı zamanda, bu ders öğrencilere bir yazılımın mimarisini çeşitli teknikler kullanarak nasıl tanımlayabileceklerini ve üzerinde hangi işlemleri gerçekleştirebileceklerini öğretmeyi hedeflemektedir	This course covers the fundamental concepts of software architecture, architectural styles, methods for modeling software architecture, software architecture description languages, and the XCD software architecture description language. Additionally, it encompasses topics such as visualizing software architecture, the UML software design language, transforming software architecture into program code, defining software architecture based on non-functional system properties, and developing domain-specific software architecture. The aim of this course is to introduce students to software architecture in terms of its conceptual understanding, principles, and techniques. At the same time, this course aims to teach students how to define the architecture of a software using various techniques and perform operations on it.
İçeriği / Content	Bu ders, yazılım mimarisinin temel kavramını, mimari stillerini, yazılım mimarisini modelleme yöntemlerini, yazılım mimarisi tanımlama dillerini ve XCD yazılım mimarisi tanımlama dilini kapsamaktadır. Ayrıca, yazılım mimarisinin görselleştirilmesi, UML yazılım tasarım dili, yazılım mimarisinin program koduna dönüştürülmesi, yazılım mimarisinin fonksiyonel olmayan sistem özelliklerine göre tanımlanması ve alan bazlı yazılım mimarisi geliştirme konuları da dersin içeriğini oluşturmaktadır. Bu dersin amacı, öğrencilere yazılım mimarisini kavramsal, ilkeleri ve teknikleri açısından tanıtmaktır. Aynı zamanda, bu ders öğrencilere bir yazılımın mimarisini çeşitli teknikler kullanarak nasıl tanımlayabileceklerini ve üzerinde hangi işlemleri gerçekleştirebileceklerini öğretmeyi hedeflemektedir	This course covers the fundamental concepts of software architecture, architectural styles, methods for modeling software architecture, software architecture description languages, and the XCD software architecture description language. Additionally, it encompasses topics such as visualizing software architecture, the UML software design language, transforming software architecture into program code, defining software architecture based on non-functional system properties, and developing domain-specific software architecture. The aim of this course is to introduce students to software architecture in terms of its conceptual understanding, principles, and techniques. At the same time, this course aims to teach students how to define the architecture of a software using various techniques and perform operations on it.

Önerilen Diğer Hususlar / Recommended Other Considerations	yok	None
Staj Durumu / Internship Status	yok	None
Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	yok	None
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)		

ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Modüler, esnek ve ölçeklenebilir yazılım sistemlerinin tasarımını yapabilme yeteneği kazanır.	Gains the ability to design modular, flexible and scalable software systems.
2	Mikroservis mimarisi, SOA ve diğer modern mimari yaklaşımlarını anlayarak uygulama yeteneği geliştirilir.	Implementation skills are developed by understanding microservice architecture, SOA and other modern architectural approaches.
3	Yazılım kalitesi, performansı ve güvenliği için tasarım desenleri ve mimari prensipleri kullanma becerisi edinir.	Acquires the ability to use design patterns and architectural principles for software quality, performance and security.
4	Dağıtık sistemlerin mimarisini tasarlama ve uygulama yeteneği geliştirilir.	The ability to design and implement the architecture of distributed systems is developed.
5	Yazılım mimarisi kararlarını verirken iş gereksinimlerini, performansı ve güvenliği dikkate alma yetisi kazanır.	Gains the ability to consider business requirements, performance and security when making software architecture decisions.

HAFTALIK DERS İÇERİĞİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Mimari Stilleri				
	Software Architecture Styles				
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Bağlayıcıları				
	Software Connectors				
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Mimari Modellemeye Giriş				
	Introduction to Architectural Modeling				
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Mimari Modelleme				
	Architectural Modeling				
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Yazılım Mimarisi Tanımlama Dilleri				
	Software Architecture Description Languages				

6	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	XCD Mimari Tanımlama Dilleri				
	XCD Architecture Description Languages				
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	XCD Mimari Tanımlama Dilleri				
	XCD Architecture Description Languages				
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Vize				
	Midterm exam				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Mimari Modellerin Görselleştirilmesi				
	Visualization of Architectural Models				
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Mimari Modellerin Analizi				
	Analysis of Architectural Models				
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Mimari Modelden Program Koduna Geçiş				
	Transition from Architectural Model to Program Code				

12	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Uygulanmış Mimariler				
	Implemented Architectures				
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Fonksiyonel Olmayan Sistem Özelliklerine göre Yazılım Mimarisi Tasarlama				
	Designing Software Architecture based on Non-Functional System Characteristics				
14	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Alan bazlı Yazılım Mühendisliği				
	Field-based Software Engineering				

DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçi Etkinlikleri / Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Ara Sınav / Midterm Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		40

Yarıyıl (Yıl) Sonu Etkinlikleri / End Of Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Final Sınavı / Final Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		60

Etkinliklerinin Başarı Notuna Katkı Yüzdesi(%) Toplamı / Total Percentage of Contribution (%) to Success Grade:	100
Değerlendirme Tipi / Evaluation Type:	

İŞ YÜKÜ / WORKLOADS

Etkinlikler / Workloads	Sayı / Number	Süresi (Saat) / Duration (Hours)	Toplam İş Yüğü (Saat) / Total Work Load (Hour)
Ara Sınav / Midterm Examination	1	72.00	72.00
Ara Sınav İçin Bireysel Çalışma / Individual Study for Mid term Examination	1	1.00	1.00
Final Sınavı / Final Examination	1	76.00	76.00
Final Sınavı için Bireysel Çalışma / Individual Study for Final Examination	1	1.00	1.00
Toplam / Total:	4	150.00	150.00
Dersin AKTS Kredisi = Toplam İş Yüğü (Saat) / 25.00 (Saat/AKTS) = 150.00/25.00 = 6.00 ~ 6.00 / Course ECTS Credit = Total Workload (Hour) / 25.00 (Hour / ECTS) = 150.00 / 25.00 = 6.00 ~ 6.00			

PROGRAM VE ÖĞRENME ÇIKTISI / PROGRAM LEARNING OUTCOMES

Öğrenme Çıktıları / Learning Outcomes	Program Çıktıları / Program Outcomes														
	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8	1.1.9	1.1.10	1.1.11	1.1.12	1.1.13	1.1.14	1.1.15
1.Modüler, esnek ve ölçeklenebilir yazılım sistemlerinin tasarımını yapabilme yeteneği kazanır. / Gains the ability to design modular, flexible and scalable software systems.	5	5	5	4	5	5	4	5	5	4	5	5	4	5	5
2.Mikroservis mimarisi, SOA ve diğer modern mimari yaklaşımlarını anlayarak uygulama yeteneği geliştirilir. / Implementation skills are developed by understanding microservice architecture, SOA and other modern architectural approaches.	5	5	5	4	5	4	5	5	4	4	5	5	5	4	5
3.Yazılım kalitesi, performansı ve güvenliği için tasarım desenleri ve mimari prensipleri kullanma becerisi edinir. / Acquires the ability to use design patterns and architectural principles for software quality, performance and security.	5	4	5	5	5	5	4	5	5	4	5	4	5	4	5
4.Dağıtık sistemlerin mimarisini tasarlama ve uygulama yeteneği geliştirilir. / The ability to design and implement the architecture of distributed systems is developed.	5	4	4	5	5	5	4	4	5	5	5	5	5	5	5
5.Yazılım mimarisi kararlarını verirken iş gereksinimlerini, performansı ve güvenliği dikkate alma yetisi kazanır. / Gains the ability to consider business requirements, performance and security when making software architecture decisions.	4	5	5	5	5	4	5	5	4	5	5	4	5	5	4

Katkı Düzeyi / Contribution Level : 1-Çok Düşük / Very low, 2-Düşük / Low, 3-Orta / Moderate, 4-Yüksek / High, 5-Çok Yüksek / Very high