

GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	Advanced Programming / Advanced Programming	
Ders Kodu / Course Code	YZM517	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	Master without Thesis / Master without Thesis	
Ders Akts Kredi / ECTS	6.00	
Haftalık Ders Saati (Kuramsal) / Course Hours For Week (Theoretical)	3.00	
Haftalık Uygulama Saati / Course Hours For Week (Objected)	0.00	
Haftalık Laboratuar 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 dersin amacı, öğrencilere ileri seviye programlama kavramlarını ve tekniklerini öğretmek, nesne yönelimli programlama prensiplerine derinlemesine hakim olmalarını sağlamak, kalıtım, polimorfizm, veri kapsülleme, soyut sınıflar, arayüzler gibi konulara odaklanmak, istisnalar ve hata yönetimi konularında bilgi vermek, ileri düzey fonksiyonel programlama konularını işlemek, hata ayıklama teknikleri, modül kullanımı, çok iş parçacıklı ve asenkron programlama gibi ileri seviye konuları ele almak ve veritabanı entegrasyonu ile ORM gibi konulara değinmektedir.	The aim of this course is to teach students advanced programming concepts and techniques, to provide students with an in-depth understanding of object-oriented programming principles, to focus on topics such as inheritance, polymorphism, data encapsulation, abstract classes, interfaces, exceptions and error handling, to cover advanced functional programming topics, debugging techniques, module usage, multithreaded and asynchronous programming, and database integration and ORM.
İçeriği / Content	Bu dersin amacı, öğrencilere ileri seviye programlama kavramlarını ve tekniklerini öğretmek, nesne yönelimli programlama prensiplerine derinlemesine hakim olmalarını sağlamak, kalıtım, polimorfizm, veri kapsülleme, soyut sınıflar, arayüzler gibi konulara odaklanmak, istisnalar ve hata yönetimi konularında bilgi vermek, ileri düzey fonksiyonel programlama konularını işlemek, hata ayıklama teknikleri, modül kullanımı, çok iş parçacıklı ve asenkron programlama gibi ileri seviye konuları ele almak ve veritabanı entegrasyonu ile ORM gibi konulara değinmektedir.	The aim of this course is to teach students advanced programming concepts and techniques, to provide students with an in-depth understanding of object-oriented programming principles, to focus on topics such as inheritance, polymorphism, data encapsulation, abstract classes, interfaces, exceptions and error handling, to cover advanced functional programming topics, debugging techniques, module usage, multithreaded and asynchronous programming, and database integration and ORM.
Önerilen Diğer Hususlar / Recommended Other Considerations	YOK	NONE
Staj Durumu / Internship Status	-	-

Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	"Python Cookbook" - David Beazley, Brian K. Jones "Fluent Python" - Luciano Ramalho "Design Patterns: Elements of Reusable Object-Oriented Software" - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides	"Python Cookbook" - David Beazley, Brian K. Jones "Fluent Python" - Luciano Ramalho "Design Patterns: Elements of Reusable Object-Oriented Software" - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)	-	

ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Nesne yönelimli programlama prensiplerini ve ileri programlama tekniklerini anlama becerisi geliştirmek.	Developing the ability to understand object-oriented programming principles and advanced programming techniques.
2	Gelişmiş programlama dilleri, tasarım desenleri ve yazılım mimarisi konularında bilgi sahibi olma yetkinliği kazanmak.	Acquiring knowledge in advanced programming languages, design patterns, and software architecture.
3	Büyük ölçekli yazılım projelerini planlama, geliştirme ve yönetme yetkinliği geliştirmek.	Developing competence in planning, developing, and managing large-scale software projects.
4	Veritabanı entegrasyonu, ağ programlaması ve web uygulama geliştirme gibi ileri programlama konularını uygulama becerisi kazanmak.	Gaining the ability to implement advanced programming topics such as database integration, network programming, and web application development.
5	Yazılım kalitesini artırmak için test otomasyonu, hata ayıklama ve performans optimizasyonu tekniklerini öğrenmek ve uygulama yetkinliği geliştirmek.	Learning and applying test automation, debugging, and performance optimization techniques to improve software quality.

HAFTALIK DERS İÇERİĞİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Nesne Yönelimli Programlama'ya Giriş				
	Introduction to Object-Oriented Programming				
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Sınıflar ve Nesneler				
	Classes and Objects				
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Kalıtım ve Polimorfizm				
	Inheritance and Polymorphism				
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Veri Kapsülleme ve Sarmalayıcılar				
	Encapsulation and Wrappers				
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Soyut Sınıflar ve Arabirimler				
	Abstract Classes and Interfaces				

6	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	İleri Nesne Yönelimli Programlama Konuları				
	Advanced Object-Oriented Programming Topics				
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	İstisnalar ve Hata Yönetimi				
	Exceptions and Error Handling				
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Ara Sınav				
	Midtern Exam				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	İleri Fonksiyonel Programlama Konuları				
	Advanced Functional Programming Topics				
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	İleri Modül Kullanımı ve Paketleme				
	Advanced Module Usage and Packaging				
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Çok İş Parçacıklı ve Asenkron Programlama				
	Multithreaded and Asynchronous Programming				

12	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dekorâtörler ve Meta-programlama				
	Decorators and Metaprogramming				
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Veritabanı Entegrasyonu ve ORM (Object-Relational Mapping)				
	Database Integration and Object-Relational Mapping				
14	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	İleri Programlama Araçları ve Teknikler (Ör. Test Otomasyonu)				
	Advanced Programming Tools and Techniques (e.g., Test Automation)				
15	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Final Sınavı				
	Final Exam				

DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçerik 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	1.00	1.00
Ara Sınav İçin Bireysel Çalışma / Individual Study for Mid term Examination	1	76.00	76.00
Final Sınavı / Final Examination	1	1.00	1.00
Final Sınavı için Bireysel Çalışma / Individual Study for Final Examination	1	72.00	72.00
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.Nesne yönelimli programlama prensiplerini ve ileri programlama tekniklerini anlama becerisi geliştirmek. / Developing the ability to understand object-oriented programming principles and advanced programming techniques.	4	5	4	5	5	4	5	5	5	5	5	5	5	5	5
2.Gelişmiş programlama dilleri, tasarım desenleri ve yazılım mimarisi konularında bilgi sahibi olma yetkinliği kazanmak. / Acquiring knowledge in advanced programming languages, design patterns, and software architecture.	4	5	5	4	5	5	4	4	5	5	5	5	4	5	5
3.Büyük ölçekli yazılım projelerini planlama, geliştirme ve yönetme yetkinliği geliştirmek. / Developing competence in planning, developing, and managing large-scale software projects.	4	5	5	5	5	5	5	4	4	5	4	4	4	4	5
4.Veritabanı entegrasyonu, ağ programlaması ve web uygulama geliştirme gibi ileri programlama konularını uygulama becerisi kazanmak. / Gaining the ability to implement advanced programming topics such as database integration, network programming, and web application development.	5	5		4	4	4	4	5	5	4	4	4	4	5	5
5.Yazılım kalitesini artırmak için test otomasyonu, hata ayıklama ve performans optimizasyonu tekniklerini öğrenmek ve uygulama yetkinliği geliştirmek. / Learning and applying test automation, debugging, and performance optimization techniques to improve software quality.	5	5	4	4	4	4	5	4	4	4	4	5	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