GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	Design Pattens / Design Pattens	
Ders Kodu / Course Code	ESOF417	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	Bachelor / Bachelor	
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	4	
Öğretim Sistemi / Teaching System	Daytime Class / Daytime Class	
Eğitim Dili / Education Language		
Ön Koşulu Olan Ders(ler) / Precondition Courses	Yoktur.	No
Amacı / Purpose	Dersin amacı, nesneye yönelik modelleme ve tasarımda yaygın olarak karşılaşılan sorunların çözümüne yönelik tasarım kalıplarını öğretmektir.	The aim of the course is to teach object-oriented modeling and design patterns for solving common problems in design.
İçeriği / Content	NYP temel ilkelerinin ve UML şemalarının hatırlatılması, Dörtlü çete tasarım kalıpları, Seçme kod kusurları	Reminder of NYP core principles and UML schemes, Quad gang design patterns, Select code flaws
Önerilen Diğer Hususlar / Recommended Other Considerations	Yok	None
Staj Durumu / Internship Status	Yok	None
Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	Design Patterns - Elements of Reusable OO Software, Erich Gamma et.al (Gang of Four), Addison-Wesley, 1994 Refactoring: Improving the Design of Existing Code, Martin Fowler. Addison-Wesley, 1999	Design Patterns - Elements of Reusable OO Software, Erich Gamma et.al (Gang of Four), Addison-Wesley, 1994 Refactoring: Improving the Design of Existing Code, Martin Fowler. Addison-Wesley, 1999
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)	Dr. Öğr.Üyesi Aliye SARAÇ	

ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Öğrenciler literatürdeki temel tasarım kalıplarını kullanarak yazılım gerçekleştirebilme yeteneği kazanır. Öğrenciler en sık ortaya çıkan kodlama kusurlarını tanıma ve düzeltme yeteneği kazanır.	Students gain the ability to implement software using the basic design patterns in the literature. Students gain the ability to recognize and correct the most common coding defects.
2	Öğrenciler nesne yönelimli analiz, tasarım ve kodlama etkinliklerini daha yetkin düzeyde yürütebilme yeteneği kazanır. Öğrenciler yazılım bakım etkinliklerinde adım adım iyileştirme yöntemini kullanma yeteneği kazanır.	Students gain the ability to carry out object-oriented analysis, design and coding activities at a more competent level. Students gain the ability to use the step-by-step improvement method in software maintenance activities.
3	Öğrenciler bir yazılımın dışarıya verdiği servisleri değiştirmeden o yazılımı iyileştirebilme yeteneği kazanır.	Students gain the ability to develop a software without changing the outsourced services.

HAFTALIK DERS İÇERİĞİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	NYP temel ilkelerinin ve UML şemalarının hatırlatılması				
	Reminder of NYP core principles and UML schemes				
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Kod Kusurları (Duplicate code, Long method, Large Class) ve Yeniden Düzenleme Eylemleri				
	Duplicate code, Long method, Large Class				
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Kod Kusurları (Divergent Change, Shotgun Surgery, Primitive Obsession) ve Yeniden Düzenleme Eylemleri				
	Divergent Change, Shotgun Surgery, Primitive Obsession				
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	MVC, Abstract Factory, Factory Method				
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	Memento, Builder, Iterator				

6	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	Chain of Responsibility, Command, Façade				
7	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	Mediator, Observer, Flyweight				
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Ara sınav				
	Midterm exam				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	State, Singleton, Composite				
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	Decorator, Adapter				
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Dörtlü Çete tasarım kalıpları ve uygulamaları				
	Decorator, Adapter				

12	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Konu Tekrarı				
	Practice				
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Konu tekrarı				
	Practice				

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 Yükü (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	10.00	10.00
Bireysel Çalışma / Self Study	10	5.00	50.00
Derse Katılım / Attending Lectures	10	4.00	40.00
Ev Ödevi / Homework	2	15.00	30.00
Final Sınavı / Final Examination	1	1.00	1.00
Final Sınavı içiin Bireysel Çalışma / Individual Study for Final Examination	1	10.00	10.00
Toplam / Total:	26	46.00	142.00
Dersin AKTS Kredisi = Toplam İş Yükü (Saat) / 25.00 (Saat/AKTS) = 142.00/25.00 = 5.68 ~ / Course ECTS Credit = Total Workload (Hour) / 25.00 (Hour / ECTS) = 142.00 / 25.00 = 5.68 ~			

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.Öğrenciler literatürdeki temel tasarım kalıplarını kullanarak yazılım gerçekleştirebilme yeteneği kazanır. Öğrenciler en sık ortaya çıkan kodlama kusurlarını tanıma ve düzeltme yeteneği kazanır. / Students gain the ability to implement software using the basic design patterns in the literature. Students gain the ability to recognize and correct the most common coding defects.	4										5
2.Öğrenciler nesne yönelimli analiz, tasarım ve kodlama etkinliklerini daha yetkin düzeyde yürütebilme yeteneği kazanır. Öğrenciler yazılım bakım etkinliklerinde adım adım iyileştirme yöntemini kullanma yeteneği kazanır. / Students gain the ability to carry out object-oriented analysis, design and coding activities at a more competent level. Students gain the ability to use the step-by-step improvement method in software maintenance activities.		3		5							5
3.Öğrenciler bir yazılımın dışarıya verdiği servisleri değiştirmeden o yazılımı iyileştirebilme yeteneği kazanır. / Students gain the ability to develop a software without changing the outsourced services.					4						5

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