**ITU**

**DERS KATALOG FORMU**

**(Course Catalogue Form)**

| **Dersin Adı:**  Nesneye Dayalı Modelleme ve Tasarım | **Course Name:**  Object Oriented Modeling and Design |
| --- | --- |

| **Kodu (Course Code)** | **Yarıyıl (Semester)** | **Kredisi (Local Credits)** | **AKTS Kredisi (ECTS Credits)** | **Ders Uygulaması, Saat/Hafta** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Ders (Theoretical)** | **Uygulama (Tutorial/Recitation)** | **Laboratuvar (Laboratory)** |
| BLG468E | 8 | 2 | 4 | 2 | 0 | 0 |

| **Bölüm/Program**  **(Department/Program)** | Bilgisayar Mühendisliği / Computer Engineering |
| --- | --- |

| **Dersin Türü**  **(Course Type)** | Mühendislik Tasarım  (Engineering Design) | **Dersin Dili (Course Language)** | İngilizce  English/ |
| --- | --- | --- | --- |
| **Ders Zorunluluğu (Course Compulsion)** | | Seçmeli (Elective) | |

| **Dersin Önkoşulları (Course Prerequisites)** | BLG 252/E Object-Oriented Programming | | | |
| --- | --- | --- | --- | --- |
| **Dersin Mesleki Bileşene Yüzde Katkısı**  **(Course Category by Content Percentage)** | Temel Bilim  (Basic Science) | Temel Mühendislik (Engineering Science) | Mühendislik Tasarım (Engineering Design) | İnsan ve Toplum Bilim (General Education) |
| 0% | 20% | 80% | 0% |

| **Dersin İçeriği (Course Description)** | Esnek ve tekrar kullanılabilir yazılım bileşenleri oluşturmak için kullanılan nesneye dayalı tasarım prensipleri ve kalıpları. Tasarımın yanı sıra ders kapsamında nesneye dayalı yazılım geliştirmenin diğer temel konularına da değinilecektir. Bu konular; tümleştirilmiş yazılım geliştirme süreci (UP), kullanım senaryoları ve problem uzayının modellemesidir. Tüm modeller UML kullanılarak ifade edilecektir. |
| --- | --- |
| Principles and patterns of object oriented design that are used to create flexible and reusable software modules. This course also covers some basic topics of object-oriented software development such as unified process (UP), use cases, and domain modeling. Analysis and design models will be presented using the UML. |
| **Dersin Amacı (Course Objective)** | 1. Tasarlanacak olan sistemin paydaşlarının gereksinimlerinin yönlendirdiği bir süreci izlemenin önemini öğretmek. 2. Tasarım kalıplarını ve bu kalıpların dayandığı nesneye dayalı kavramları öğretmek. 3. Esnek ve tekrar kullanılabilir yazılım bileşenleri tasarlamayı öğretmek. 4. Tasarım kalıplarının, gerçek yazılım problemlerinin çözümünü sağlamak için nasıl kullanılacağını öğretmek. |
| 1. Teaching the importance of following a process that is driven by the requirements of the stakeholders of the system. 2. Teaching design patterns and their underlying object-oriented concepts. 3. Teaching how to design flexible and reusable software components. 4. Teaching how to use design patterns to provide solutions to real world software design problems. |
| **Dersin Öğrenme Çıktıları (Course Learning Outcomes)** | 1. Yazılım dünyasındaki sorunları ve neden iyi bir tasarıma gerek duyulduğunu bilme. 2. Gereksinimleri belirleyebilme ve kullanım senaryolarını yazabilme 3. Problem uzayını çözümlemeyi ve modellemeyi bilme 4. Nesneye dayalı tasarım prensiplerini, örneğin “uzman”, “az bağımlılık”, “yüksek uyum”, “dolaylılık” ve “çok şekillilik” kavrama 5. Nesneye dayalı tasarım prensiplerinin ve kalıplarının dayandığı esneklik, tekrar kullanılabilirlik, kolay anlaşılabilirlik gibi gerekçeleri bilme 6. Önemli GoF tasarım kalıplarını bilme 7. Gerçek yazılım problemlerini çözmek üzere tasarım kalıplarını kullanabilme |
| 1. Knowing the problems in the software industry and being aware of the need for a high-quality design 2. Ability to perform requirement analysis and write use-cases 3. Ability to analyze the problem domain and create the domain model 4. Knowing important object oriented design principles such as “expert”, “low coupling”, “high cohesion”, “indirection”, and “polymorphism” 5. Knowing the reasoning for object-oriented design principles and patterns such as flexibility, reusability, and understandability 6. Knowing important GoF design patterns. 7. Ability to use design patterns to solve real world software design problems |

| **Ders Kitabı (Textbook)** | 1. Craig Larman, Applying UML and Patterns, An Introduction to OOA/D and Iterative Development, 3/e, Prentice Hall PTR, 2005. 2. Eric Freeman, Elisabeth Robson, Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software, 2/e, O’REILLY, 2020. |
| --- | --- |
| **Diğer Kaynaklar (Other References)** | Gamma E., Helm R., Johnson R., Vlissides J., Design Patterns : Elements of Reusable Object-Oriented Software, Reading MA, Addison-Wesley, 1995. |

| **Ödevler ve Projeler (Homeworks & Projects)** | Dönem içinde beş adet çözümleme ve tasarım ödevi verilir. |
| --- | --- |
| Five analysis/design assignments are given |
| **Laboratuvar Uygulamaları (Laboratory Work)** | - |
| - |
| **Bilgisayar Kullanımı (Computer Use)** | - |
| - |
| **Diğer Uygulamalar (Other Activities)** | - |
| - |

| **Başarı Değerlendirme Sistemi**  **(Assessment Criteria)** | **Faaliyetler (Activities)** | **Adedi (Quantity)** | **Değerlendirmedeki Yüzde Katkısı**  **(Effects on Grading by Percentage)** |
| --- | --- | --- | --- |
| **Yıl İçi Sınavları (Midterm Exams)** | 1 | 40% |
| **Kısa Sınavlar (Quizzes)** | - | - |
| **Ödevler (Homework)** | 5 | 20% |
| **Projeler (Projects)** | - | - |
| **Dönem Ödevi/Projesi (Term Paper/Project)** | - | - |
| **Laboratuvar Uygulaması (Laboratory Work)** | - | - |
| **Diğer Uygulamalar (Other Activities)** | - | - |
| **Final Sınavı (Final Exam)** | 1 | 40% |

**DERS PLANI**

**(Course Plan)**

| **Hafta** | **Konu** | **Dersin Çıktıları** |
| --- | --- | --- |
| **1** | Giriş: Yazılım dünyasındaki sorunlar ve dersin ana konuları | 1 |
| **2** | Gereksinimlerin analizi, kullanım senaryoları | 2 |
| **3** | Analiz, problem uzayının modellenmesi | 3 |
| **4** | Larman (GRASP) tasarım prensipleri: Denetçi, Yaratıcı, Uzman, Az bağımlılık, iyi uyum, | 4 |
| **5** | GRASP tasarım prensipleri ile tasarım örnekleri | 4,5,7 |
| **6** | Larman (GRASP) tasarım prensipleri: Çok şekillilik, “Yapay sınıf”, “Dolaylılık”, “Değişimlerden korunma” | 4 |
| **7** | GRASP tasarım prensipleri ile tasarım örnekleri | 4,5,7 |
| **8** | GoF Tasarım kalıplarına giriş. Adaptör, Fabrika, Tekil Nesne | 5,6 |
| **9** | GoF tasarım kalıplarına devam. Strateji, Bileşik nesne | 5,6 |
| **10** | GoF tasarım kalıplarına devam. Cephe kalıbı, değişken işletme kuralları problemi, Gözlemci kalıbı | 6,7 |
| **11** | GoF tasarım kalıpları ile tasarım örnekleri | 5,6,7 |
| **12** | GoF tasarım kalıplarına devam. Dekoratör, Şablon metot | 5,6 |
| **13** | GoF tasarım kalıplarına devam. Köprü kalıbı | 5,6 |
| **14** | GoF tasarım kalıpları ile tasarım örnekleri | 5,6,7 |

| **Week** | **Topic** | **Course Outcome** |
| --- | --- | --- |
| **1** | Introduction. Problems in the software industry, main topics of the course | 1 |
| **2** | Requirement Analysis, Use Cases | 2 |
| **3** | Analysis, domain modeling | 3 |
| **4** | Larman (GRASP) Design Principles: Controller, Creator, Expert, Low coupling, High cohesion | 4 |
| **5** | Design examples using Larman (GRASP) Design Principles | 4,5,7 |
| **6** | Larman (GRASP) Design Principles: Polymorphism, Pure Fabrication, Indirection, Protected variations | 4 |
| **7** | Design examples using Larman (GRASP) Design Principles | 4,5,7 |
| **8** | Introduction to GoF design patterns. Adapter, Factory, Singleton | 5,6 |
| **9** | GoF design patterns. Strategy, Composite | 5,6 |
| **10** | GoF design patterns. Facade, pluggable design rules, Observer | 6,7 |
| **11** | Design examples using GoF design patterns | 5,6,7 |
| **12** | GoF design patterns. Decorator, Template method | 5,6 |
| **13** | GoF design patterns. Bridge | 5,6 |
| **14** | Design examples using GoF design patterns | 5,6,7 |

**DERSİN BİLGİSAYAR MÜHENDİSLİĞİ ÖĞRENCİ ÇIKTILARI İLE İLİŞKİSİ**

**Relationship between the Course and Student Outcomes**

**(1: “Little”, 2: “Partial”, 3: “Full”, Leave blank if your answer is “None”)**

| **Computer Engineering Department Program Outcomes and Performance Criteria** | | **Level of Contribution** | | |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** |
| 1 | an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics |  | X |  |
| 2 | an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors |  |  | X |
| 3 | an ability to communicate effectively with a range of audiences |  |  |  |
| 4 | an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts |  |  |  |
| 5 | an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives |  |  |  |
| 6 | an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions |  |  |  |
| 7 | an ability to acquire and apply new knowledge as needed, using appropriate learning strategies |  |  |  |

**HAZIRLANMA BİLGİSİ**

**Edition Information**

| **Prepared by** | **Date** | **Signature** |
| --- | --- | --- |
| **Dr.Feza BUZLUCA** | **6.12.2020** |  |
| **Approved by** | **Date** | **Signature** |
| **Dr. Tolga Ovatman** | **07.12.2020** |  |