**ITU**

**DERS KATALOG FORMU**

**(Course Catalogue Form)**

| **Dersin Adı:**  Veri Yapıları | **Course Name:**  Data Structures |
| --- | --- |

| **Kodu (Course Code)** | **Yarıyıl (Semester)** | **Kredisi (Local Credits)** | **AKTS Kredisi (ECTS Credits)** | **Ders Uygulaması, Saat/Hafta** | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Ders (Theoretical)** | **Uygulama (Tutorial/Recitation)** | **Laboratuvar (Laboratory)** |
| BLG223E | 3 | 3.5 | 7 | 3 | 1 | 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)** | | Zorunlu (Compulsory) | |

| **Dersin Önkoşulları (Course Prerequisites)** | BLG102/E Intr. to Sci.&Eng. Comp. (C)  or  BIL104/E Intr. to Programming in C  or  BIL105/E Intr. to Sci.&Eng. Comp. (C) | | | |
| --- | --- | --- | --- | --- |
| **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) |
| 15% | 35% | 50% | - |

| **Dersin İçeriği (Course Description)** | Ders temel veri yapıları (yığın, kuyruk, liste, ağaç, ikili sıralama ağacı, heap) ve bu yapılara ilişkin algoritmaları inceleyecektir. |
| --- | --- |
| The course involves the study of basic data structures (eg, stack, queue, list, tree, binary search tree, heap) and associated algorithms. |
| **Dersin Amacı (Course Objective)** | 1. Temel veri yapılarını ve bu yapıları işlemekte kullanılan algoritmaları tanıtmak. 2. Öğrencilere, problemlere uygun veri yapılarını belirleyebilmek için gerekli olan teorik ve pratik bilgi birikimi sağlamak. 3. Öğrencilere, farklı algoritmaların etkinliğini karşılaştırabilmeleri için gerekli olan teorik ve pratik bilgi birikimi sağlamak. |
| 1. To introduce basic data structures and algorithms for manipulating them. 2. To provide students with theoretical and practical knowledge to analyse the data structure needs of particular problems. 3. To provide students with theoretical and practical knowledge to compare the efficiency of various algorithms. |
| **Dersin Öğrenme Çıktıları (Course Learning Outcomes)** | 1. Öğrenciler, algoritma tasarımı temelleri öğreneceklerdir. 2. Öğrenciler temel veri yapıları ve uygulamalarını öğrenip, gerçekleyebileceklerdir. 3. Öğrenciler dinamik veri yapılarının temellerini, uygulamalarını ve gerçeklemelerini öğreneceklerdir. 4. Öğrenciler, işaretçi tipi ve özyineleme gibi ileri programlama kavramlarını kullanarak problemler çözebileceklerdir. 5. Öğrenciler, mühendislik uygulamaları için, C dilinde programlar tasarlama, gerçekleme, test etme ve hata ayıklamayı öğreneceklerdir. |
| 1. The student will learn the fundamentals of algorithm design. 2. Students will be able to identify and implement basic data structures and their applications. 3. The student will learn the principles, applications, and implementations of dynamic data structures. 4. Students will be able to solve problems using advanced programming concepts like pointers and recursion. 5. The student will learn to design, implement, test, and debug computer programs using C++ for engineering applications. |

| **Ders Kitabı (Textbook)** | Ders Notları /Lecture Notes |
| --- | --- |
| **Diğer Kaynaklar (Other References)** | Deitel & Deitel, C++ How to program, 6th edition, Deitel and Assoc. Inc., 2008 |

| **Ödevler ve Projeler (Homeworks & Projects)** | 3 ödev verilecektir, ödevlerin süresi ikişer hafta olacaktır |
| --- | --- |
| 3 Homeworks will be given. Each homework will take two weeks to complete |
| **Laboratuvar Uygulamaları (Laboratory Work)** | - |
| - |
| **Bilgisayar Kullanımı (Computer Use)** | Tüm ödevler C++ programlama ortamının var olduğu bilgisayarlar üzerinde yapılacaktır. |
| All homeworks and projects are performed on computers with C++ programming environments. |
| **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 | 30% |
| **Kısa Sınavlar (Quizzes)** | - | - |
| **Ödevler (Homework)** | 3 | 30% |
| **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, C/C++’da dosya işlemleri, derse giriş örneği: Telefon defteri | 1,2 |
| **2** | C’de işaretçiler ve diziler | 2,4 |
| **3** | Fonksiyon çağrıları, dinamik bellek, soyut veri tipi | 2,3,5 |
| **4** | Bağlantılı liste  (Lab: Ders örneğinde listeler üzerinde güncelleme işlemi, güncelleme ve aramanın performansının arttırılması) | 2,3 |
| **5** | Bağlantılı liste türleri, liste uygulamaları | 1,2,3,5 |
| **6** | Yığın | 1,2,3 |
| **7** | Kuyruk | 1,2,3 |
| **8** | Yığın ve kuyruk uygulamaları | 1,2,3,5 |
| **9** | Özyinelemeli programlama | 1,4 |
| **10** | Özyinelemeli programlama uygulamaları | 1,4,5 |
| **11** | Ağaç | 1,2,3 |
| **12** | İkili sıralama ağacı | 1,2,3 |
| **13** | Ağaç uygulamaları | 1,2,3,5 |
| **14** | Hazır veri yapısı kütüphaneleri | 4,5 |

| **Week** | **Topic** | **Course Outcome** |
| --- | --- | --- |
| **1** | Introduction, file operations in C/C++, introduction to course example: Phone-book | 1,2 |
| **2** | Pointers and arrays in C | 2,4 |
| **3** | Function calls, dynamic memory, abstract data type | 2,3,5 |
| **4** | Linked list | 2,3 |
| **5** | Types of linked list, linked list applications | 1,2,3,5 |
| **6** | Stack | 1,2,3 |
| **7** | Queue | 1,2,3 |
| **8** | Applications of stack and queue | 1,2,3,5 |
| **9** | Recursive programming | 1,4 |
| **10** | Recursive programming applications | 1,4,5 |
| **11** | Tree | 1,2,3 |
| **12** | Binary sort tree | 1,2,3 |
| **13** | Tree applications | 1,2,3,5 |
| **14** | Standard template library | 4,5 |

**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 |  |  | X |
| 7 | an ability to acquire and apply new knowledge as needed, using appropriate learning strategies |  |  |  |

**HAZIRLANMA BİLGİSİ**

**Edition Information**

| **Prepared by** | **Date** | **Signature** |
| --- | --- | --- |
| **Dr.Tolga Ovatman** | **24.11.2020** |  |
| **Approved by** | **Date** | **Signature** |
| **Dr.Tolga Ovatman** | **24.11.2020** |  |