



Erzurum Atatürk University
Faculty of Engineering
Computer Engineering Department
CEN 104: Object Oriented Programming - Spring 2021

PROGRAMMING PROJECT 2
(Due Date: May 25th, 2021 - 23:59)

In your second programming project, you are expected to write a simple console application (in C++ programming language) that can be used as a library automation program in universities.

The details of the implementation of object-oriented programming using the concepts of class, object and inheritance are listed below:

1. There will be two main classes:
 - a. **User**
 - b. **Book**
2. There will be two types of users (**Admin**, **Member**) and the classes you will define for both users must be inherited from the base class named **User**.
3. **User** class must consist of name-surname, ID, account name and password member variables and functions that will set (Set) and return (Get) the values of these variables. (You have the freedom to add other member variables and member functions you will need to the class definition for both this class and other classes.)
4. One of the member functions of the **User** base class should be a **bool** function that will verify the name and password of the account they enter in order for users to access the system.
5. The class defined as **Admin** that is derived from the **User** class should contain the following member functions related to the administration of the system:
 - a. **Add Member:** This function is used to add a new member to the system. Before adding members, the account name entered for the new member must be checked. Having more than one member with the same account name in the system should be prevented.
 - b. **Delete Member:** This function ensures that a member defined in the system is deleted from the system.
 - c. **Add Book:** This function allows a new book to be added to the system.
 - d. **Delete Book:** This function ensures that a book whose Id is entered is deleted from the system.
 - e. **List Members:** This function prints all information (values of member variables) of all members defined in the system.
 - f. **List Books:** This function prints the information of the books registered in the system on the screen.

6. The **Member** class to be derived from the **User** class has a member variable named **memberType** that is of string data type, and 3 different values can be assigned to this member variable:
 - a. Student
 - b. Personnel
 - c. Faculty Member.
7. The class named **Member** derived from the **User** class must contain the following member functions related to the actions that members can perform:
 - a. **Find Book:** Searches among the books registered in the system by entering the author or book name, returns the information of the searched book or books and presses on the screen.
 - b. **Borrow Book:** If the book whose ID is entered is not borrowed by another user, the book is borrowed by this member. Lending is done by changing the status of the related book. If the member is a student, he / she can borrow a maximum of 5 books, staff can borrow 3 books and the instructor 10 books.
 - c. **Return a Book:** A user can return the book for which he has entered his ID into the system. This is done by changing the status of the book. Members can only return books borrowed by themselves.
 - d. **List Books:** Displays the information about the books registered in the system.
8. **Book** class includes properties such as book title, Id, author, year of publication and Get and Set functions belonging to these properties. At the same time, the status of the book must be stored in a member variable. If the book has been borrowed by a member, its status should be "Hold", if it has not been borrowed, its status should be "Unhold". If the book is in "Hold" status, the information on who borrowed the book should also be kept. (For example, the id of the user who borrowed the book can be used for this purpose.)

Programming Details

1. Assume that there is only one **Admin** in the system and create a single object of the **Admin** class in **main**. After that, update the relevant member functions and the username of this object as **admin** and the password to **1234**.
2. **Do not use** STL or any other dynamic data structure in your project.
3. Keep the books and the members registered in the system in static arrays. For this purpose, define static arrays in main that will store 100 books (for example **Book bk[100]**) and 100 members (for example, **Member mbr[100]**). In all transactions related to books and/or members, these arrays should be transferred to the relevant member functions as parameters.
4. When the program is run, the **main** function must first print the following **main menu**. (Main menu and sub-menus should be implemented with a **switch-case** programming structure.)

```

ANA MENU

1 Admin Girisi
2 Uye Girisi
3 Cikis

Seciminizi Girin:█
```

Eng: Main Menu / 1. Admin Login / 2. Member Login / 3. Exit / Enter Your Choise

5. For **Admin** login, 1 should be pressed and then the username and password of the **Admin** user should be entered from the keyboard. If the entered username and password are correct, the following Admin menu should be printed:

```
1 Admin Girisi
2 Uye Girisi
3 Cikis

Seciminizi Girin:1
Admin Kullanici Adi:admin
Sifre:1234

Sifre Kabul edildi
ADMIN MENUSU

1 Kitap Ekle
2 Kitap Sil
3 Kitaplari Listele
4 Uye Ekle
5 Uyeleri Listele
6 Uyeleri Sil
7 Cikis

Seciminizi Girin:
```

Eng: 1. Admin Login / 2. Member Login / 3. Exit / Enter Your Choise: 1

Admin User Name: admin

Password: 1234

Password accepted

Admin Menu / 1 Add Book / 2 Delete Book / 3 List Books / 4 Add Member / 5 List Members / 6 Delete Member / 7 Exit

Enter Your Choise:

6. At this stage, press 1 to add a new book to the system. Enter the information about the book on the keyboard and if there is no book registered in the system with the same author and book name, perform the book adding as in the screen output below. Automatically assign an Id (integers starting from 0) to each added book. At the end of the process, go back to the **Admin menu**.

```
Seciminizi Girin:1
Kitap adi:C++
Yazar adi:A.James
Kitap Eklendi

ADMIN MENUSU

1 Kitap Ekle
2 Kitap Sil
3 Kitaplari Listele
4 Uye Ekle
5 Uyeleri Listele
6 Uyeleri Sil
7 Cikis

Seciminizi Girin:
```

Eng: Enter Your Choise:1

Book Name:C++ / Author Name:A.Jones / Book Added

Admin Menu / 1 Add Book / 2 Delete Book / 3 List Books / 4 Add Member / 5 List Members / 6 Delete Member / 7 Exit

Enter Your Choise:

7. To delete a book stored in the system, select the action number 2 in the Admin menu. Enter the book's Id from the keyboard and delete the related book from the array where you have kept the books. You can do this by sliding the array to the left. At the end of the process, go back to the **Admin menu**.
8. Select action number 3 from Admin menu to display the books stored in the system as below. At the end of the process, go back to the **Admin menu**.

ID	Kitap Adi:	Yazar Adi	Statu
0	C++	J.Adams	Unhold
1	C	S.David	Unhold
2	Python	S.Aslan	Unhold
3	C++	S.David	Unhold
4	C#	S.David	Unhold
5	.Net	M.James	Unhold

Eng: ID / Book Name / Author Name / Status

9. In order to add a new member to the system, select the action number 4 in the **Admin** menu. As in the screenshot below, take the member name, account name and member type from the keyboard and add it to the related array element as an object. Member type must be "Student", "Personnel" or "Faculty Member ". It should be checked before adding that there is not more than one member with the same account name in the system. The value of the first password of each member created by the admin is set as "temp" by default (the relevant member should be able to change this password later). At the end of the process, go back to the Admin menu.

```

Seciminizi Girin:4
Uye bilgilerini girin
ad-soyad:A.Turk
Hesap Adi:Alia
Uye Tipi:Ogrenci
Bu hesap isminde bir uye kayitli. Tekrar deneyin

```

Eng: Enter Your Choise:4

Enter Member Info / name-surname:A.Turk / Account Name:Alia / Member Type:Student

A member is registered with this account name. Try again.

10. In order to display all members registered in the system on the screen, select the action numbered 5 in the Admin menu as in the screenshot below. At the end of the process, go back to the Admin menu.

1	Ayşe Tekin	Oğrenci AyşeT	
2	Aysel Sayın	Personel	AyselS
3	Ali Aslan	OğretimElemanı	AliA
4	Ali Türk	Oğrenci aTürk	
5	Pelin Mutlu	OğretimElemanı	p.mutlu
6	Derin Can	Oğrenci derin.can	

11. To delete a registered member in the system, select the action number 6 in the **Admin** menu. For deletion, take the account name information from the screen. You can delete the element by sliding the array to the left where you keep the members. At the end of the process, go back to the Admin menu.
12. To exit the **Admin** menu and return to the main menu, select the action number 7 as in the screenshot below.

```

ADMIN MENUSU
1 Kitap Ekle
2 Kitap Sil
3 Kitapları Listele
4 Üye Ekle
5 Üyeleri Listele
6 Üyeleri Sil
7 Çıkış

Seciminizi Girin:7

1 Admin Girişi
2 Üye Girişi
3 Çıkış

Seciminizi Girin:

```

Eng: Admin Menu / 1 Add Book / 2 Delete Book / 3 List Books / 4 Add Member / 5 List Members / 6 Delete Member / 7 Exit

Enter Your Choice:7

1. Admin Login / 2. Member Login / 3. Exit / Enter Your Choice: 2.

13. Select action number 2 from the main menu for member actions. As can be seen in the screen output below, the accuracy of the account name and password must be checked before entering the member menu. After both the account name and password are verified, the member can perform the actions. (When a new **Member** is created by the **admin**, the member's first password is always set as "temp", but this password can be changed in the **Member Operations**.)

```
Seciminizi Girin:2
Uye Kullanici Adi:p.mutlu
Sifre:temp

Sifre Kabul edildi
UYE MENUSU

1 Kitap Ara
2 Kitap Odunc Al
3 Kitap Iade Et
4 Kitaplari Listele
5 Uyelik Islemleri
6 Cikis

Seciminizi Girin:
```

Eng: Enter Your Choise:2

Member User Name:p.metlu / Password:temp / Password accepted

Member Menu / 1. Find Book / 2. Borrow book / 3. Return Book / 4. List Books / 5. Member Operations / 6. Exit

Enter Your Choise:

14. A **Member** defined in the system can search for a book in the system with the title of the book and the name of the author. For this, as shown in the screenshot below, action number 1 should be selected from the **Member** menu. The books found as a result of the search should be printed on the screen. At the end of the process, one should return to the Member menu.

```
Seciminizi Girin:1
Kitap Adi girin:C++
Yazar Adi girin:*

aradiginiz kitap bulundu. Kitap bilgileri:
0      J.Adams C++      Unhold
3      S.David C++      Unhold

UYE MENUSU

1 Kitap Ara
2 Kitap Odunc Al
3 Kitap Iade Et
4 Kitaplari Listele
5 Uyelik Islemleri
6 Cikis

Seciminizi Girin:
```

Eng: Enter Your Choise:1

Enter book name:C++ / Enter author name:* / Book you search for is found. Book Info:

....

Member Menu / 1. Find Book / 2. Borrow book / 3. Return Book / 4. List Books / 5. Member Operations / 6. Exit

Enter Your Choise:

15. As can be seen in the screen output below, action number 2 must be selected from the **Member** menu in order to borrow a book. To borrow books, the book ID must be entered and in the meantime, the maximum number of books that the member can borrow, depending on the member type, must not be exceeded. The status of the borrowed book should be updated to "Hold". At the end, one should return to the Member menu.

```
Seciminizi Girin:2
Kitap Id:0
Kitap baska bir uye tarafindan alinmis
UYE MENUSU
1 Kitap Ara
2 Kitap Odunc Al
3 Kitap Iade Et
4 Kitaplari Listele
5 Uyelik Islemleri
6 Cikis
Seciminizi Girin:█
```

Eng: Enter Your Choise:2

Book Id:0 / The book has been borrowed by another member

Member Menu / 1. Find Book / 2. Borrow book / 3. Return Book / 4. List Books / 5. Member Operations / 6. Exit

Enter Your Choise:

16. In order to return books, action number 3 must be selected from the member menu. The id of the book to be returned must be entered on the keyboard and the status of the book must be changed to "Unhold". If the book whose Id is entered is not in "Hold" status or is not borrowed by the member who made the transaction, an error message should be printed on the screen. At the end of the process, one should return to the Member menu.
17. As can be seen in the screen output below, action 4 in the Member menu should be selected in order to list all the books registered in the system. At the end of the process, one should return to the Member menu.

0	C++	J.Adams	Hold
1	C	S.David	Unhold
2	Python	S.Aslan	Unhold
3	C++	S.David	Unhold
4	C#	S.David	Unhold
5	.Net	M.James	Unhold

18. A member must select action number 5 from the Member menu to update his password by entering a new password. At the end of the process, one should return to the Member menu.
19. You should return to the Main Menu with the action number 6 in the Member menu.
20. It should be possible to log out of the system with the operation number 3 in the Main Menu and thus the program should be terminated.

RULES and GUIDELINES

- 1) You are expected to write your application using a compiler.
- 2) Define a class named **Book_YourStudentNumber**. For this, create the two files below and add them to your project.
 - i. **Book_YourStudentNumber.h** (Definition file of the class)
 - ii. **Book_YourStudentNumber.cpp** (Application file of the class)
- 3) Define a class named **User_YourStudentNumber**. For this, create the two files below and add them to your project.
 - i. **User_YourStudentNumber.h** (Definition file of the class)
 - ii. **User_YourStudentNumber.cpp** (Application file of the class)
- 4) Define a class named **Book_YourStudentNumber**. For this, create the two files below and add them to your project.
 - i. **Member_YourStudentNumber.h** (Definition file of the class)
 - ii. **Member_YourStudentNumber.cpp** (Application file of the class)
- 5) Define a class named **Book_YourStudentNumber**. For this, create the two files below and add them to your project.
 - i. **Admin_YourStudentNumber.h** (Definition file of the class)
 - ii. **Admin_YourStudentNumber.cpp** (Application file of the class)
- 6) Create the file **Executive.cpp**, which contains the main function, and add it to your project.
- 7) You must name the folder and compress it as **FirstnameLastname_StudentNumber_CEN104_Project2.zip** and upload it to **DBS** (Ders Bilgi Sistemi) until the deadline. The compressed folder will contain **only** the following **9 files**.
 - i. **Book_YourStudentNumber.h**
 - ii. **Book_YourStudentNumber.cpp**
 - iii. **User_YourStudentNumber.h**
 - iv. **User_YourStudentNumber.cpp**
 - v. **Member_YourStudentNumber.h**
 - vi. **Member_YourStudentNumber.cpp**
 - vii. **Admin_YourStudentNumber.h**
 - viii. **Admin_YourStudentNumber.cpp**
 - ix. **Executive.cpp**
- 8) At the top of each **.h** and/or **.cpp** file in your project, there should be a section called **General Explanations about the Application** and this section shall contain
 - your name,
 - your lastname,
 - your student number,
 - your department,
 - name and code of the course,
 - academic semester,
 - project number and
 - name of the Lecturer shall be **given in comment lines**.
- 9) It is **not allowed** to use global variables. (Variables defined before **main** are called global variables. In other words, variables whose scope are both **main** functions and other functions are called global variables.)
- 10) C style library declaration is **not allowed**. All library declarations must be in C++ style. For example, **<ctime>** should be used instead of **<time.h>**.

- 11) Relevant libraries must be included in your application. For example, in order to use the **rand** and **srand** functions, the **<cstdlib>** library must be included to your program.
 - 12) Camel notation should be used in variable naming and pascal notation should be used in function naming.
 - 13) Any use of recursive functions is **not allowed**.
 - 14) Changing the program flow using the **goto** statement is **not allowed**.
 - 15) Comment lines **must** be added to the program where necessary.
 - 16) The writing style of the development environment should not be interfered with. Tabs should be used properly.
 - 17) The project you will submit should **ONLY** be the product of your own efforts. Do not forget that very serious sanctions await those who act on the contrary.
 - 18) Do not forget that all codes submitted by students in **A, B, C and English teaching groups** will be tested with a special software for similarities.
 - 19) Please note that projects that do not comply with the project delivery rules and/or contain plagiarism will not be evaluated.
- *Turkish description is provided below.

İkinci programlama projenizde üniversitelerde kütüphane otomasyon programı olarak kullanılabilecek basit bir konsol uygulamasını C++ programlama dili ile yazmanız beklenmektedir.

Nesneye yönelik programlamanın sınıf, nesne ve kalıtım kavramlarını kullanarak gerçekleyeceğiniz uygulamanın detayları aşağıda listelenmiştir:

1. Uygulama 2 temel sınıftan oluşmaktadır:
 - a. **Kullanıcı**
 - b. **Kitap**
2. Uygulamada iki tip kullanıcı (**Admin (Yönetici)**, **Üye**) vardır ve her iki kullanıcı için tanımlayacağınız sınıflar, **Kullanıcı** isimli temel sınıftan kalıtım yolu ile oluşturulmalıdır.
3. **Kullanıcı** sınıfı ad-soyad, Id, hesap adı ve şifre üye değişkenlerinden ve bu değişkenlerin değerlerini belirleyecek (Set) ve geri döndürecek (Get) fonksiyonlardan oluşmalıdır. (İhtiyacınız olacak diğer üye değişkenleri ve üye fonksiyonları hem bu sınıf hem de diğer sınıflar açısından sınıf tanımına ekleme özgürlüğüne sahipsiniz.)
4. **Kullanıcı** temel sınıfının üye fonksiyonlarından biri, kullanıcıların sistemde işlem yapabilmeleri için girdikleri hesabın adını ve şifreyi doğrulayacak bir bool fonksiyon olmalıdır.
5. **Admin** kullanıcısı için tanımlanacak ve **Kullanıcı** sınıfından türetilmiş sınıf, sistemin yönetilmesi ile ilgili aşağıdaki üye fonksiyonları içermelidir:
 - a. **Üye Ekleme:** Bu fonksiyon, sisteme yeni bir üye eklemek için kullanılır. Üye ekleme işlemi yapılmadan önce, yeni üye için girilen hesap adı kontrol edilmelidir. Sistemde aynı hesap adına sahip birden fazla üye olması engellenmelidir.
 - b. **Üye Silme:** Bu fonksiyon, sistemde tanımlı bir üyenin sistemden silinmesini sağlar.
 - c. **Kitap Ekleme:** Bu fonksiyon, sisteme yeni bir kitap eklenmesini sağlar.
 - d. **Kitap Silme:** Bu fonksiyon, Id'si girilen bir kitabın sistemden silinmesini sağlar.
 - e. **Üyeleri Listeleme:** Bu fonksiyon, sistemde tanımlı bütün üyelerin tüm bilgilerini (üye değişkenlerin değerlerini) ekrana basar.
 - f. **Kitap Listeleme:** Bu fonksiyon, sistemde kayıtlı kitaplara ait bilgileri ekrana basar.
6. **Kullanıcı** sınıfından türetililecek **Üye** sınıfının **uyeTipi** isimli ve **string** veri tipinde bir üye değişkeni mevcuttur ve bu üye değişkene 3 farklı değer atanabilmektedir:
 - a. Öğrenci
 - b. Personel
 - c. Öğretim Elemanı.
7. **Üye** isimli ve **Kullanıcı** sınıfından türetilmiş sınıf, üyelerin yapabileceği işlemler ile ilgili aşağıdaki üye fonksiyonları içermelidir:
 - a. **Kitap Arama:** Yazar veya kitap adı girilerek sisteme kayıtlı kitaplar arasında arama yapar, aranan kitap veya kitapların bilgilerini geriye döndürür ve ekrana basar.
 - b. **Kitap Ödünç Alma:** Id'si girilen kitap başka bir kullanıcı tarafından ödünç alınmamışsa, kitap bu üyeye ödünç verilir. Ödünç verme işlemi ilgili kitabın statüsü değiştirilerek yapılır. Bir üye eğer öğrenci ise mevcut durumda en fazla 5 kitap, personel ise 3 kitap ve öğretim elemanı ise 10 kitap ödünç alabilir.
 - c. **Kitap İade Etme:** Bir kullanıcı Id'sini girdiği kitabı sisteme iade edebilir. Bu işlem kitabın statüsü değiştirilerek yapılır. Üyeler sadece kendi ödünç aldığı kitapları iade edebilir.
 - d. **Kitap Listeleme:** Sistemde kayıtlı kitaplar ile ilgili bilgileri ekrana basar.
8. **Kitap** sınıfı, kitap adı, Id, yazar, basım yılı gibi özellikleri ve bu özelliklere ait Get ve Set fonksiyonlarını içerir. Aynı zamanda kitabın statüsü de bir üye değişkeni içerisinde saklanmalıdır. Eğer kitap bir üye tarafından ödünç alınmış durumdaysa statüsü "Hold", ödünç alınmamış durumdaysa statüsü "Unhold" olmalıdır. Kitap "Hold" statüsünde ise

kitabın kim tarafından ödünç alındığı bilgisi de saklanmalıdır. (Örneğin kitabı ödünç alan kullanıcının id'si bu amaçla kullanılabilir.)

Programlama Ayrıntıları

1. Sistemde sadece bir tane **Admin** olduğunu varsayınız ve **main** içerisinde **Admin** sınıfının bir tek nesnesini oluşturunuz. Sonrasında ilgili üye fonksiyonları ile bu nesnenin kullanıcı adını **admin** ve şifresini de **1234** olarak güncelleyiniz.
2. Projenizde STL veya başka bir dinamik veri yapısı kullanmayınız.
3. Kütüphanedeki kitapları ve sisteme kayıtlı üyeleri statik diziler içerisinde saklayınız. Bu amaçla **main** içerisinde 100 kitap (örneğin **Kitap ktp[100]**) ve 100 üye (örneğin **Uye uy[100]**) saklayacak statik diziler tanımlayınız. Kitaplarla ve/veya üyelerle alakalı tüm işlemlerde ilgili üye fonksiyonlara bu diziler parametre olarak aktarılmalıdır ve gerekli güncellemeler yapılmalıdır.
4. Program çalıştırıldığında **main** fonksiyonu öncelikle aşağıdaki ana menüyü ekrana basmalıdır. (Ana menü ve alt menüler **switch-case** programlama yapısı ile gerçekleştirilmelidir.)

```
ANA MENU
1 Admin Girişi
2 Uye Girişi
3 Cikis
Seciminizi Girin:█
```

5. **Admin** girişi için 1'e basılmalı ve daha sonra **Admin** kullanıcısının kullanıcı adı ve şifresi klavye üzerinden alınmalıdır. Girilen kullanıcı adı ve şifre doğru ise aşağıdaki **Admin** menüsü ekrana basılmalıdır:

```
1 Admin Girisi
2 Uye Girisi
3 Cikis

Seciminizi Girin:1
Admin Kullanici Adi:admin
Sifre:1234

Sifre Kabul edildi
ADMIN MENUSU

1 Kitap Ekle
2 Kitap Sil
3 Kitaplari Listele
4 Uye Ekle
5 Uyeleri Listele
6 Uyeleri Sil
7 Cikis

Seciminizi Girin:
```

6. Bu aşamada sisteme yeni bir kitap eklemek için 1'e basınız. Kitap ile ilgili bilgileri klavyeden giriniz ve aynı yazar ve kitap adında sistemde kayıtlı bir kitap yoksa, kitap ekleme işini aşağıdaki ekran çıktısındaki gibi gerçekleştiriniz. Her eklenen kitaba otomatik olarak bir Id (0'dan başlayan tam sayılar) atayınız. İşlem bitiminde **Admin** menüsüne geri dönünüz.

```
Seciminizi Girin:1
Kitap adi:C++
Yazar adi:A.James
Kitap Eklendi

ADMIN MENUSU

1 Kitap Ekle
2 Kitap Sil
3 Kitaplari Listele
4 Uye Ekle
5 Uyeleri Listele
6 Uyeleri Sil
7 Cikis

Seciminizi Girin:
```

7. Sistemde kayıtlı bir kitabı silmek için **Admin** menüsünde 2 numaralı işlemi seçiniz. Kitabın Id'sini klavyeden alınız ve ilgili kitabı içerisinde kitapları sakladığınız diziden siliniz. Bu işlemi diziye bir sola kaydırarak yapabilirsiniz. İşlem bitiminde **Admin** menüsüne geri dönünüz.

8. Sistemde kayıtlı kitapları aşağıdaki gibi ekrana basmak için **Admin** menüsünde 3 numaralı işlemi seçiniz. İşlem bitiminde **Admin** menüsüne geri dönünüz.

ID	Kitap Adı:	Yazar Adı	Statu
0	C++	J.Adams	Unhold
1	C	S.David	Unhold
2	Python	S.Aslan	Unhold
3	C++	S.David	Unhold
4	C#	S.David	Unhold
5	.Net	M.James	Unhold

9. Sisteme yeni bir üye eklemek için **Admin** menüsünde 4 numaralı işlemi seçiniz. Aşağıdaki ekran çıktısındaki gibi üye adını, hesap ismini ve üye tipini klavyeden alarak ilgili dizi elemanına bir nesne olarak ekleyiniz. Üye tipi, “Ogrenci”, “Personel” veya “OgretimElemani” olmalıdır. Sistemde aynı hesap ismine sahip birden fazla üye olmadığı ekleme işleminden önce kontrol edilmelidir. **Admin** tarafından oluşturulan her üyenin ilk şifresinin değeri varsayılan (default) “temp” olarak belirlenir (ilgili üye bu şifresini sonradan değiştirebilmelidir.). İşlem bitiminde **Admin** menüsüne geri dönünüz.

```
Seciminizi Girin:4
Uye bilgilerini girin
ad-soyad:A.Turk
Hesap Adı:AliA
Uye Tipi:Ogrenci
Bu hesap isminde bir uye kayıtlı. Tekrar deneyin
```

10. Sistemde kayıtlı bütün üyeleri ekrana basmak için aşağıdaki ekran çıktısında olduğu gibi **Admin** menüsünde 5 numaralı işlemi seçiniz. İşlem bitiminde **Admin** menüsüne geri dönünüz.

1	Ayşe Tekin	Ogrenci	AyşeT
2	Aysel Sayın	Personel	AyselS
3	Ali Aslan	OgretimElemani	AliA
4	Ali Turk	Ogrenci	aTurk
5	Pelin Mutlu	OgretimElemani	p.mutlu
6	Derin Can	Ogrenci	derin.can

11. Sistemde kayıtlı bir üyeyi silmek için **Admin** menüsündeki 6 numaralı işlemi seçiniz. Silme işlemi için ekrandan hesap adı bilgisini alınız. Silme işlemini üyeleri sakladığınız diziyi bir sola kaydırarak yapabilirsiniz. İşlem bitiminde **Admin** menüsüne geri dönünüz.

12. **Admin** menüsünden çıkıp ana menüye dönmek için aşağıdaki ekran çıktısında olduğu gibi 7 numaralı işlemi seçiniz.

```
ADMIN MENUSU
1 Kitap Ekle
2 Kitap Sil
3 Kitaplari Listele
4 Uye Ekle
5 Uyeleri Listele
6 Uyeleri Sil
7 Cikis

Seciminizi Girin:7

1 Admin Girisi
2 Uye Girisi
3 Cikis

Seciminizi Girin:█
```

13. Üye işlemleri için ana menüden 2 numaralı işlemi seçiniz. Aşağıdaki ekran çıktısında görüldüğü üzere üye işlemleri menüsüne girilmeden önce hesap adı ve şifre doğruluğu kontrol edilmelidir. Hesap adının ve şifrenin her ikisi doğrulandıktan sonra üye işlemleri gerçekleştirmelidir. (**Admin** tarafından yeni bir **Uye** oluşturulduğunda üyenin ilk şifresi her zaman “temp” olarak belirlenir, ama bu şifre Üyelik İşlemleri içinde değiştirilebilir.)

```
Seciminizi Girin:2
Uye Kullanici Adi:p.mutlu
Sifre:temp

Sifre Kabul edildi
UYE MENUSU
1 Kitap Ara
2 Kitap Odunc Al
3 Kitap Iade Et
4 Kitaplari Listele
5 Uyelik Islemleri
6 Cikis

Seciminizi Girin:█
```

14. Sistemde tanımlı bir Üye, kitap adı ve yazar adı ile sistemde kitap arayabilir. Bu işlem için aşağıdaki ekran çıktısında görüldüğü üzere Üye menüsünden 1 numaralı işlem seçilmelidir.

Arama işleminin sonucunda bulunan kitaplar ekrana basılmalıdır. İşlem sonunda Üye menüsüne geri dönmelidir.

```
Seciminizi Girin:1
Kitap Adi girin:C++
Yazar Adi girin:*

aradiginiz kitap bulundu. Kitap bilgileri:
0      J.Adams C++      Unhold
3      S.David C++      Unhold

UYE MENUSU
1 Kitap Ara
2 Kitap Odunc Al
3 Kitap Iade Et
4 Kitaplari Listele
5 Uyelik Islemleri
6 Cikis

Seciminizi Girin:
```

15. Aşağıdaki ekran çıktısında görüldüğü üzere bir kitabı ödünç almak için Üye menüsünden 2 numaralı işlem seçilmelidir. Kitap ödünç almak için kitap Id'si girilmeli ve bu esnada üye tipine bağlı olarak üyenin ödünç alabileceği maksimum kitap sayısı aşılmamalıdır. Ödünç alınan kitabın statüsü "Hold" olarak güncellenmelidir. İşlem sonunda Üye menüsüne geri dönmelidir.

```
Seciminizi Girin:2
Kitap Id:0

Kitap baska bir uye tarafından alinmis

UYE MENUSU
1 Kitap Ara
2 Kitap Odunc Al
3 Kitap Iade Et
4 Kitaplari Listele
5 Uyelik Islemleri
6 Cikis

Seciminizi Girin:█
```

16. Kitap iade işlemi için üye menüsünden 3 numaralı işlem seçilmelidir. Klavyeden iade edilecek kitabın Id'si girilmeli ve kitabın statüsü ilgili üye değişkeni aracılığıyla "Unhold"

yapılmalıdır. Eğer Id'si girilen kitap "Hold" statüsünde değilse veya işlemi yapan kullanıcı tarafından ödünç alınmış durumda değilse ekrana bu durumla alakalı bir hata mesajı yansıtılmalıdır. İşlem sonunda Üye menüsüne geri dönmelidir.

17. Aşağıdaki ekran çıktısında görüldüğü üzere sistemde kayıtlı kitapların hepsini birden ekranda listelemek için Üye menüsündeki 4 numaralı işlem seçilmelidir. İşlem sonunda Üye menüsüne geri dönmelidir.

0	C++	J.Adams	Hold
1	C	S.David	Unhold
2	Python	S.Aslan	Unhold
3	C++	S.David	Unhold
4	C#	S.David	Unhold
5	.Net	M.James	Unhold

18. Bir üye, şifresini güncellemek için Üye menüsünden 5 numaralı işlemi seçmeli ve yeni şifresini belirlemelidir. İşlem sonunda Üye menüsüne geri dönmelidir.
19. Üye menüsündeki 6 numaralı işlem ile Ana Menüye dönmelidir.
20. Ana Menüdeki 3 numaralı işlem ile sistemden çıkış yapılabilmesi ve böylece program sonlandırılmalıdır.

GENEL HUSUSLAR

- 1) Uygulamanızı bir derleyici kullanarak yazmanız beklenmektedir.
- 2) **Kitap_OgrenciNumaraniz** adında bir sınıf tanımlayınız. Bu amaçla aşağıdaki iki dosyayı oluşturarak projenize ekleyiniz.
 - i. **Kitap_OgrenciNumaraniz.h** (Sınıfınızın tanım dosyası)
 - ii. **Kitap_OgrenciNumaraniz.cpp** (Sınıfınızın uygulama dosyası)
- 3) **Kullanici_OgrenciNumaraniz** adında bir sınıf tanımlayınız. Bu amaçla aşağıdaki iki dosyayı oluşturarak projenize ekleyiniz.
 - i. **Kullanici_OgrenciNumaraniz.h** (Sınıfınızın tanım dosyası)
 - ii. **Kullanici_OgrenciNumaraniz.cpp** (Sınıfınızın uygulama dosyası)
- 4) **Uye_OgrenciNumaraniz** adında bir sınıfı **Kullanici** sınıfından türeterek tanımlayınız. Bu amaçla aşağıdaki iki dosyayı oluşturarak projenize ekleyiniz.
 - i. **Uye_OgrenciNumaraniz.h** (Sınıfınızın tanım dosyası)
 - ii. **Uye_OgrenciNumaraniz.cpp** (Sınıfınızın uygulama dosyası)
- 5) **Admin_OgrenciNumaraniz** adında bir sınıfı **Kullanici** sınıfından türeterek tanımlayınız. Bu amaçla aşağıdaki iki dosyayı oluşturarak projenize ekleyiniz.
 - i. **Admin_OgrenciNumaraniz.h** (Sınıfınızın tanım dosyası)
 - ii. **Admin_OgrenciNumaraniz.cpp** (Sınıfınızın uygulama dosyası)
- 6) İçerisinde **main** fonksiyonu yer alan **Yurutucu.cpp** isimli dosyayı oluşturunuz ve projenize ekleyiniz.
- 7) **AdSoyad_OgrenciNumarasi_BM104_Proje2.zip** şeklinde isimlendireceğiniz sıkıştırılmış bir klasörün içerisine sadece aşağıda listelenen 9 dosyayı koyunuz ve son teslim saatine kadar DBS'ye (Ders Bilgi Sistemi) yükleyiniz.
 - i. **Kitap_OgrenciNumaraniz.h**
 - ii. **Kitap_OgrenciNumaraniz.cpp**

- iii. **Kullanici_OgrenciNumaraniz.h**
- iv. **Kullanici_OgrenciNumaraniz.cpp**
- v. **Uye_OgrenciNumaraniz.h**
- vi. **Uye_OgrenciNumaraniz.cpp**
- vii. **Admin_OgrenciNumaraniz.h**
- viii. **Admin_OgrenciNumaraniz.cpp**
- ix. **Yurutucu.cpp**

- 8) Projenizde yer alan her bir **.h** ve/veya **.cpp** dosyasının en üstünde **Uygulama ile İlgili Genel Açıklamalar** adında bir bölüm olmalıdır ve bu bölümde
adınız,
soyadınız,
öğrenci numaranız,
bölümünüz,
dersin adı ve kodu,
dersin verildiği akademik dönem,
projenin numarası ve
dersi veren öğretim üyesinin adı **açıklama satırı şeklinde yer almalıdır.**
- 9) Global değişken kullanımına izin **yoktur.** (**main** öncesinde tanımlanan değişkenlere global değişken denir. Başka bir deyişle ömrü/faaliyet alanı/sıkobu hem **main** fonksiyonu hem de diğer fonksiyonlar olan değişkenler global değişken olarak adlandırılırlar.)
- 10) C sitil kütüphane deklarasyonuna izin **yoktur.** Bütün kütüphane deklarasyonları C++ sitilinde olmalıdır. Örneğin **<time.h>** yerine **<ctime>** kullanılmalıdır.
- 11) Uygulamanızda ilgili kütüphaneler muhakkak **include** edilmelidir. Örneğin **rand** ve **srand** fonksiyonlarını kullanabilmek için programınıza **<cstdlib>** kütüphanesi eklenmelidir.
- 12) Değişken isimlendirmelerinde **deve notasyonu** ve fonksiyon isimlendirmelerinde **paskal notasyonu** kullanılmalıdır.
- 13) Herhangi bir rekürsif fonksiyon kullanımına izin **yoktur.**
- 14) **goto** deyimi kullanılarak program akışının değiştirilmesine izin **yoktur.**
- 15) Yazdığınız programın içerisine **MUHAKKAK** açıklayıcı satırlar eklenmelidir.
- 16) Derleyicinin yazım sitiline müdahale edilmemelidir. **Girintilere** özellikle dikkat edilmelidir.
- 17) Teslim edeceğiniz proje **SADECE** ve **SADECE** kendi emeğinizin ürünü olmalıdır. Aksine hareket edenleri çok ciddi yaptırımların beklediğini unutmayınız.
- 18) **A, B, C** ve **İngilizce öğretim** grubundaki öğrencilerin teslim ettikleri kodların tamamının özel bir yazılım ile benzerlik testine tabi tutulacağını unutmayınız.
- 19) Proje teslim kurallarına uymayan ve/veya intihal içeren projelerin değerlendirmeye alınmayacağını unutmayınız.