Object Oriented Programming

Lab Manual 4

Introduction

After a week of rigorous coding, Welcome back!

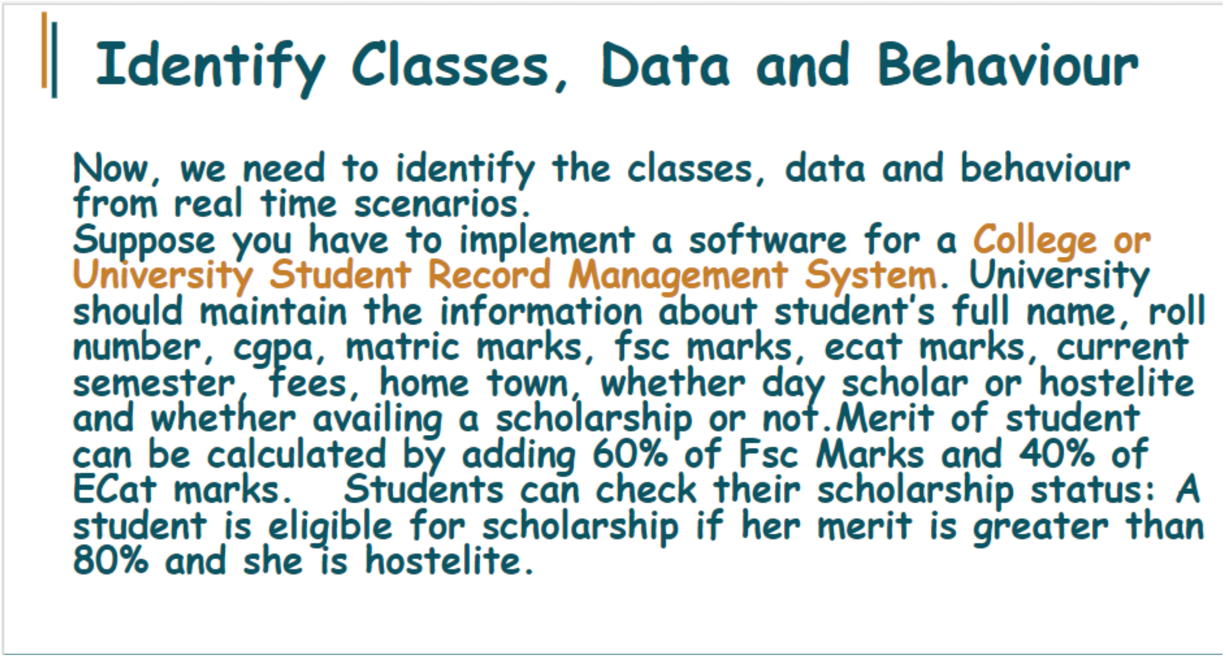
You have learned all about the Classes, Constructors, and member functions in the previous lab manuals. Let's move on to the next, new, and interesting concepts.

Students, In Object-Oriented Programming, the Class is a combination of data members and member functions. In this Lab, we will learn about including **multiple classes** into our program to achieve the object's oriented philosophy.

Let's do some coding.

Students, Recall this question from the class that you attempted and developed CRC cards.

Problem Statement 01:



Identify Classes, Data and Behaviour

Now, we need to identify the classes, data and behaviour from real time scenarios.
Suppose you have to implement a software for a College or University Student Record Management System. University should maintain the information about student's full name, roll number, cgpa, matric marks, fsc marks, ecat marks, current semester, fees, home town, whether day scholar or hostelite and whether availing a scholarship or not.Merit of student can be calculated by adding 60% of Fsc Marks and 40% of ECat marks. Students can check their scholarship status: A student is eligible for scholarship if her merit is greater than 80% and she is hostelite.

Solution

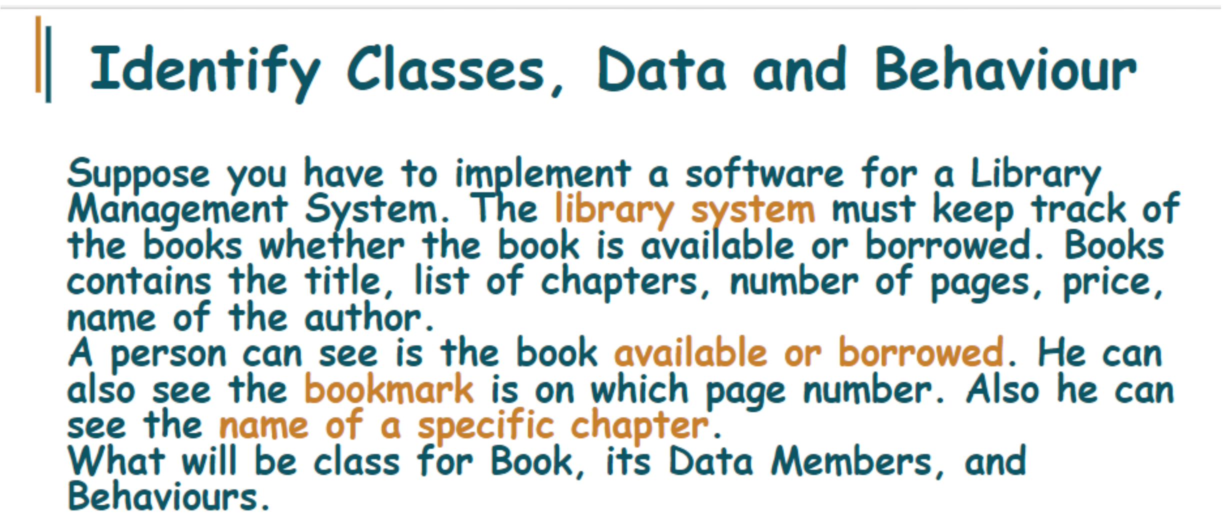
– **CRC Card**

Student
name: string rollNumber: int cgpa: float matricMarks: int fscMarks: int ecatMarks: int homeTown: string isHostelite: bool isTakingScholarship: bool
claculateMerit(): float isEligibleforScholarship(float meritPercentage): bool

Self Assessment Task: Implement the Class Student by using the CRC card that you developed in the class and use the driver program to test your member functions.

Congratulations !!!!!!!!!!!!!!!! you have implemented your first class by using the CRC card.

Problem Statement 02:



Identify Classes, Data and Behaviour

Suppose you have to implement a software for a Library Management System. The library system must keep track of the books whether the book is available or borrowed. Books contains the title, list of chapters, number of pages, price, name of the author.
A person can see is the book available or borrowed. He can also see the bookmark is on which page number. Also he can see the name of a specific chapter.
What will be class for Book, its Data Members, and Behaviours.

Solution

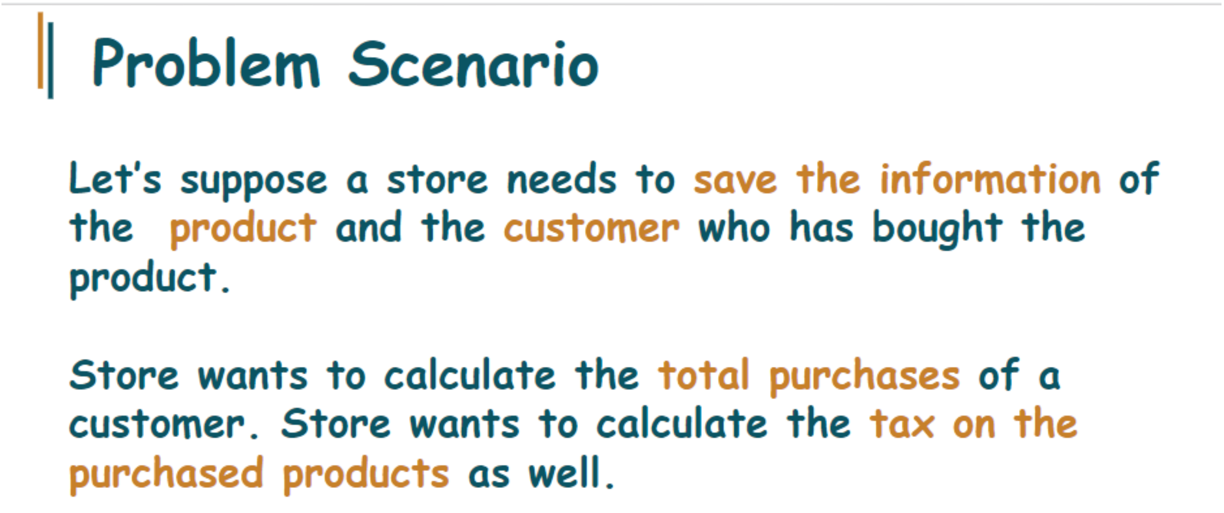
CRC Card

Book
title: string author: string pages: int chapters: List<string> bookMark: int price: int isAvailable: bool
Book(parametrized) isBookAvailable(): bool getChapter(chapterNumber: int): string getBookMark(): int

Self Assessment Task: Implement the Class Student by using the CRC card that you developed in the class and use the driver program to test your member functions.

Great Work Students !!! You guys are doing an excellent job. Well done. Take a two-minute break. Well deserved.

Problem Statement 03:



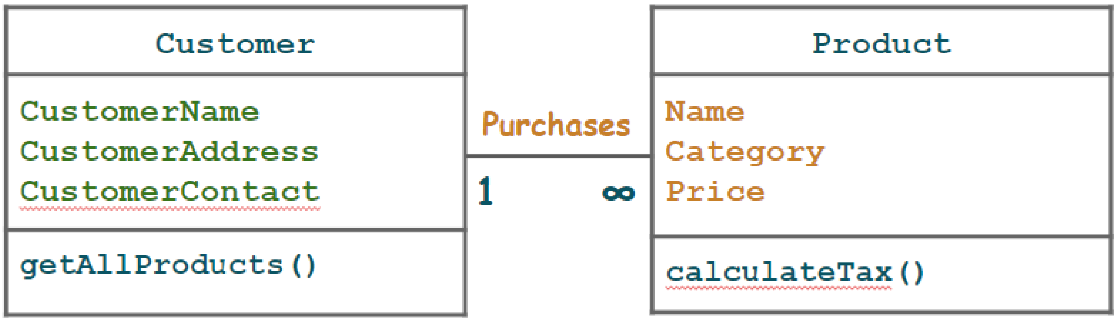
Problem Scenario

Let's suppose a store needs to save the information of the product and the customer who has bought the product.

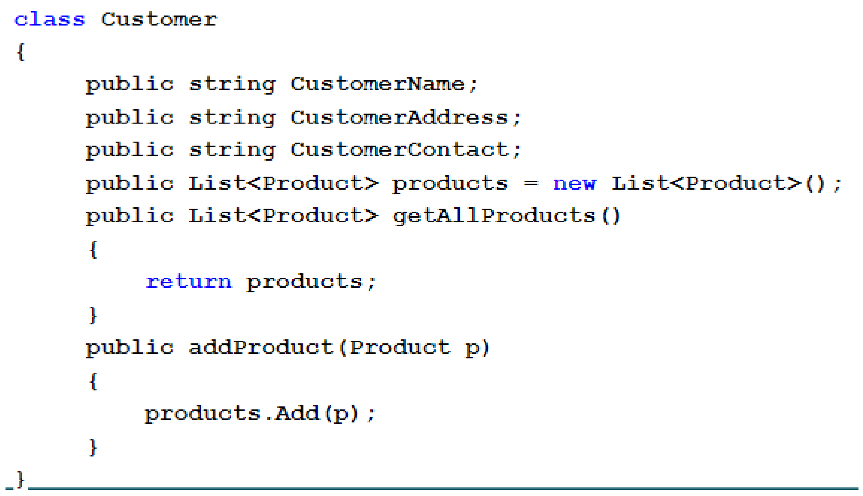
Store wants to calculate the total purchases of a customer. Store wants to calculate the tax on the purchased products as well.

Solution

CRC Card

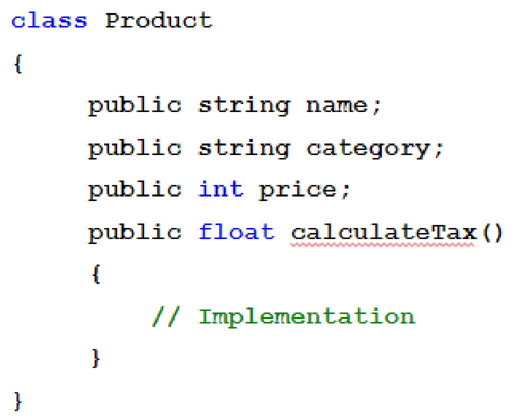


We need to define **two separate classes** in order to achieve the object-oriented philosophy. So same kinds of data shall reside in the same place.



```
class Customer
{
      public string CustomerName;
      public string CustomerAddress;
      public string CustomerContact;
      public List<Product> products = new List<Product>();
      public List<Product> getAllProducts()
      {
          return products;
      }
      public addProduct(Product p)
      {
          products.Add(p);
      }
}
```

This code creates the **Customer** class **Data Members**, **Constructor**, and **Member Function** for adding products to the list.



```
class Product
{
      public string name;
      public string category;
      public int price;
      public float calculateTax()
      {
          // Implementation
      }
}
```

This code creates the **Product** class **Data Members** and **Member Function** for calculating tax.

Self Assessment Task 01: Implement this program by using the CRC card that we have developed and then use the driver program to test your member functions.
Self Assessment Task 02: Implement the member function in the appropriate class for calculating the tax of all the products that a customer has purchased.

Challenge # 01:

Some of the characteristics of a book are the **title, author(s), publisher, ISBN, price, stock and year of publication**.

Design a class named **Book** that defines the book.

1. **Attributes:** Each object of the class Book can hold the following information about a book: title, up to four authors, publisher, ISBN, price, and number of copies in stock. To keep track of the number of authors, add another member variable.
2. **Member Functions:** Include the member functions to perform the various operations on objects of type Book. For example, the usual operations that can be performed on the title are to show the title, set the title, and check whether a title is the same as the actual title of the book. Similarly, the typical operations that can be performed on the number of copies in stock are to show the number of copies in stock, set the number of copies in stock, update the number of copies in stock, and return the number of copies in stock. Add similar operations for the publisher, ISBN, book price, and authors. Add the appropriate constructors
3. **Driver Program:** Write a menu driven program that uses the class Book and tests various operations on the objects of the class Book. Declare an array of 100 components of type Book. Some of the operations that you should perform are to add a book, search for a book by its title, search by ISBN, and update the number of copies of a book.

Challenge # 02:

In this exercise, you will design a class **Member**.

1. **Attributes:** Each object of the Member can hold the name of a **person, member ID, List of books bought, number of books bought, moneyInBank and amount spent**.
2. **Member Functions:** Include the member functions to perform the various operations on the objects of the Member—for example, modify, set, and show a person’s name. Similarly, update, modify, and show the number of books bought and the amount spent. Add the appropriate constructors.
3. **Driver Program:** Write a menu driven program to test various operations of your class Member.

Challenge # 03:

4. Using the classes designed in Programming Challenges 1 and 2, write a program to simulate a bookstore. The bookstore has two types of customers: those who are members of the bookstore and those who buy books from the bookstore only occasionally (for non-Member people store the memberID attribute as 0). Each member has to pay a $10 yearly membership fee and receives a 5% discount on each book purchased. For each member, the bookstore keeps track of the number of books purchased and the total amount spent. For every eleventh book that a member buys, the bookstore takes the average of the total amount of the last 10 books

purchased, applies this amount as a discount, and then resets the total amount spent to 0. Your program should contain a menu that gives the user different choices to effectively run the program; in other words, your program should be user driven.

a. **Add a Book:**
   i. Gather information for a new book (title, authors, publisher, ISBN, price, stock, year of publication).
   ii. Add the book to the bookstore's inventory (i.e., List of Books in the Main).
b. **Search for a Book by Title:**
   i. Enter a book title to search for.
   ii. Display information about the book if found.
c. **Search for a Book by ISBN:**
   i. Enter an ISBN to search for.
   ii. Display information about the book if found.
d. **Update Stock of a Book:**
   i. Enter the title or ISBN of the book to update.
   ii. Provide an option to increase or decrease the stock of the book.
e. **Add a Member:**
   i. Gather information for a new member (name, member ID).
   ii. Add the member to the bookstore's list (i.e., List of Members in the Main).
f. **Search for a Member by Name or ID:**
   i. Enter the name or ID of a member to search for. In Case of non-member only search based on the name.
   ii. Display information about the member if found.
g. **Update Member Information:**
   i. Enter the name or ID of a member to update. In Case of a non-member, only enter the name.
   ii. Provide options to modify the member's name, ID, or both.
h. **Purchase a Book:**
   i. Enter the name and member ID or specify as non-member (0) for occasional buyers.
   ii. Select a book to purchase, and enter the quantity.
   iii. Apply discounts for members and process the purchase.
i. **Display Total Sales and Membership Stats:**
   i. Display the total sales made by the bookstore.

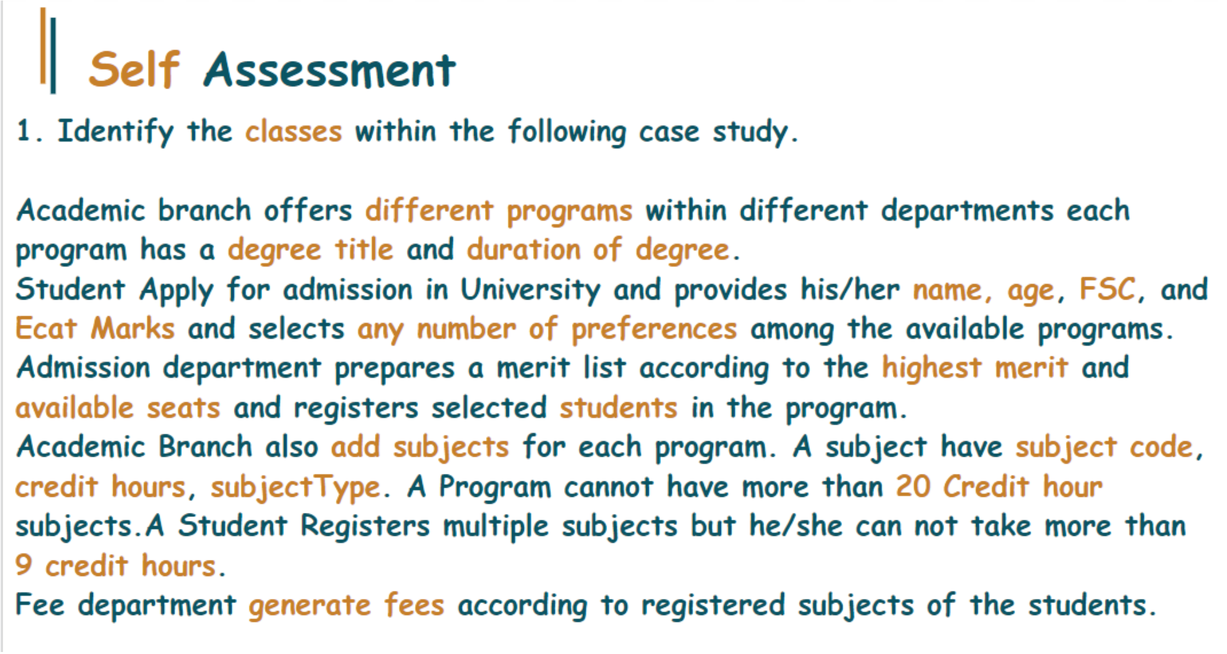
ii. Show the number of members and the total membership fee collected.

j. Exit:

i. Terminate the program.

Challenge # 04:

Read the following question carefully.



Self Assessment

1. Identify the classes within the following case study.

Academic branch offers different programs within different departments each program has a degree title and duration of degree.
Student Apply for admission in University and provides his/her name, age, FSC, and Ecat Marks and selects any number of preferences among the available programs.
Admission department prepares a merit list according to the highest merit and available seats and registers selected students in the program.
Academic Branch also add subjects for each program. A subject have subject code, credit hours, subjectType. A Program cannot have more than 20 Credit hour subjects.A Student Registers multiple subjects but he/she can not take more than 9 credit hours.
Fee department generate fees according to registered subjects of the students.

Try out yourself.

Identification of Classes

By looking at the above-mentioned self-assessment you can extract the following possible class-like structures from the given statement.

- Student
- Subject
- Degree Program

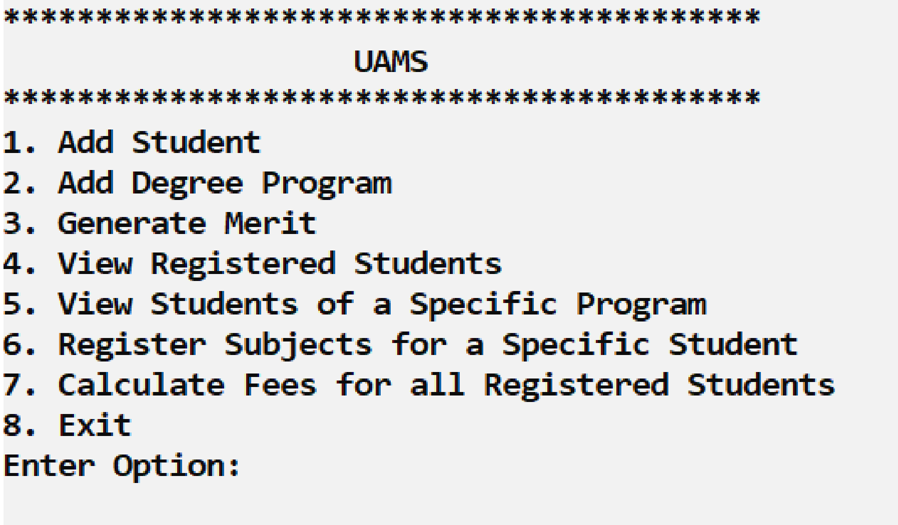
Note: Create a separate class in the same BL(Business Logic) folder of your program.

Now Try to Build the Class Diagram/Domain Model of these classes.

After creating your Class diagram you have to write the Driver Program by using your Classes.

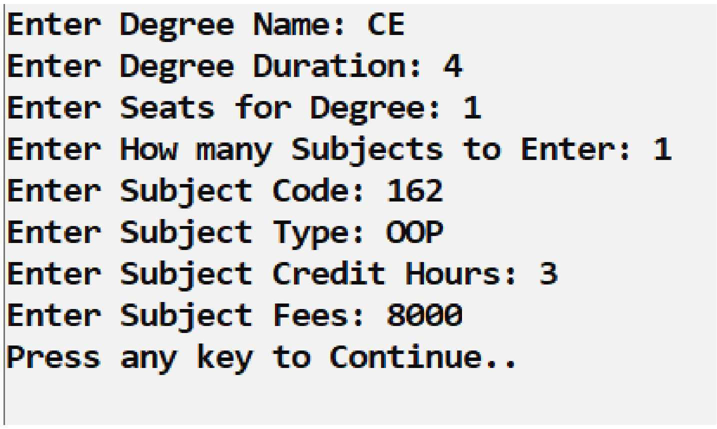
Wireframes of Driver Program:

1. Main Menu



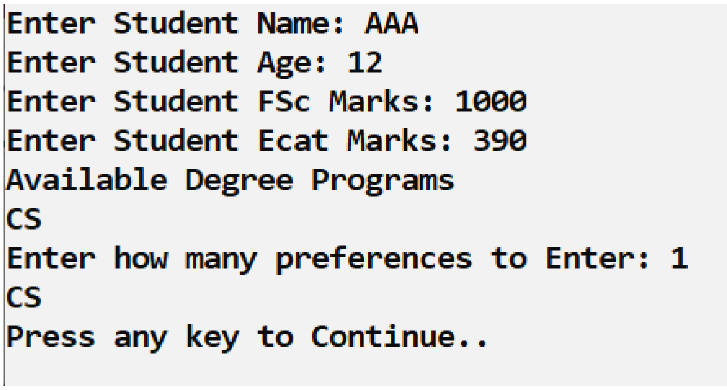
```
*****************************************
                   UAMS
*****************************************
1. Add Student
2. Add Degree Program
3. Generate Merit
4. View Registered Students
5. View Students of a Specific Program
6. Register Subjects for a Specific Student
7. Calculate Fees for all Registered Students
8. Exit
Enter Option:
```

2. Option 2: Degree Program



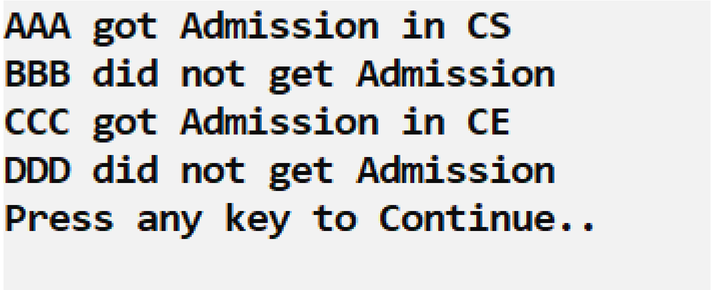
```
Enter Degree Name: CE
Enter Degree Duration: 4
Enter Seats for Degree: 1
Enter How many Subjects to Enter: 1
Enter Subject Code: 162
Enter Subject Type: OOP
Enter Subject Credit Hours: 3
Enter Subject Fees: 8000
Press any key to Continue..
```

3. Option 1: Add Student



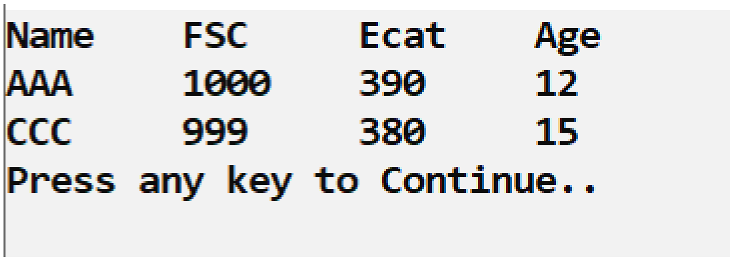
```
Enter Student Name: AAA
Enter Student Age: 12
Enter Student FSc Marks: 1000
Enter Student Ecat Marks: 390
Available Degree Programs
CS
Enter how many preferences to Enter: 1
CS
Press any key to Continue..
```

4. Option 3: Generate Merit



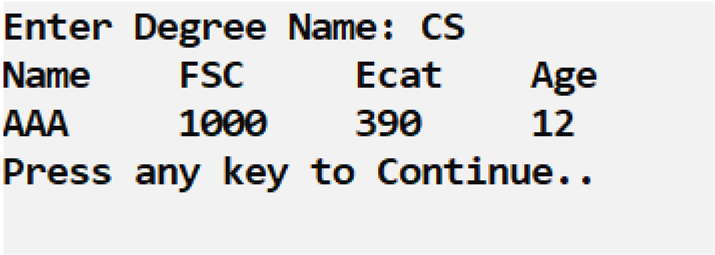
```
AAA got Admission in CS
BBB did not get Admission
CCC got Admission in CE
DDD did not get Admission
Press any key to Continue..
```

5. Option 4: Registered Students



```
Name     FSC      Ecat     Age
AAA      1000     390      12
CCC      999      380      15
Press any key to Continue..
```

6. Option 5: Specific Degree



```
Enter Degree Name: CS
Name     FSC      Ecat     Age
AAA      1000     390      12
Press any key to Continue..
```

7. Option 6: Register Subjects
   Ask the Student name and then ask for the subject code. If the conditions are satisfied then the student's subject should be registered.

8. Option 7: Generate Fee
   Fees should be generated for all the registered students

You have made it through all that. Excellent work students !!!
You guys are successfully en route to be Kamyab Programmers.

“No more Work for Today.”

Good Luck and Best Wishes !!
Happy Coding ahead :)