



First Version of UAMS System



Requirements

We want to make a Simple console based Menu System like this.

```
*****
*                               *
*           Console Based Menu System           *
*                               *
*****

Select one of the following options number...
1. Enter Option 1
2. Enter Option 2
3. Enter Option 3
Your Option..
```

Requirements

```
*****
*                               *
*       Console Based Menu System       *
*****
```

Select one of the following options number...

1. Enter Option 1
2. Enter Option 2
3. Enter Option 3

Your Option..1

You have Entered Option 1

```
*****
*                               *
*       Console Based Menu System       *
*****
```

Select one of the following options number...

1. Enter Option 1
2. Enter Option 2
3. Enter Option 3

Your Option..2

You have Entered Option 2

```
*****
*                               *
*       Console Based Menu System       *
*****
```

Select one of the following options number...

1. Enter Option 1
2. Enter Option 2
3. Enter Option 3

Your Option..3

You have Entered Option 3

Requirements

How can we do that?



```
#include <iostream>
using namespace std;
main()
{
    cout << "*****" << endl;
    cout << "          Console Based Menu System          *" << endl;
    cout << "*****" << endl;
    cout << endl;
```

```
}
```

```
#include <iostream>
using namespace std;
main()
{
    cout << "*****" << endl;
    cout << "          Console Based Menu System          *" << endl;
    cout << "*****" << endl;
    cout << endl;
    int option;
    cout << "Select one of the following options number..." << endl;
    cout << "1. Enter Option 1" << endl;
    cout << "2. Enter Option 2" << endl;
    cout << "3. Enter Option 3" << endl;
    cout << "Your Option..";
    cin >> option;
}
```

```
#include <iostream>
using namespace std;
main()
{
    cout << "*****" << endl;
    cout << "          Console Based Menu System          *" << endl;
    cout << "*****" << endl;
    cout << endl;
    int option;
    cout << "Select one of the following options number..." << endl;
    cout << "1. Enter Option 1" << endl;
    cout << "2. Enter Option 2" << endl;
    cout << "3. Enter Option 3" << endl;
    cout << "Your Option..";
    cin >> option;
    if (option == 1){
        cout << "You have Entered Option 1";
    }
    if (option == 2){
        cout << "You have Entered Option 2";
    }
    if (option == 3){
        cout << "You have Entered Option 3";
    }
}
```

```

#include <iostream>
using namespace std;
main()
{
    cout << "*****" << endl;
    cout << "          Console Based Menu System          *" << endl;
    cout << "*****" << endl;
    cout << endl;
    int option;
    cout << "Select one of the following options number..." << endl;
    cout << "1. Enter Option 1" << endl;
    cout << "2. Enter Option 2" << endl;
    cout << "3. Enter Option 3" << endl;
    cout << "Your Option..";
    cin >> option;
    if (option == 1){
        cout << "You have Entered Option 1";
    }
    if (option == 2){
        cout << "You have Entered Option 2";
    }
    if (option == 3){
        cout << "You have Entered Option 3";
    }
}

```

Can we divide this code
into functions?




```

#include <iostream>
using namespace std;
main()
{
    {
        cout << "*****" << endl;
        cout << "          Console Based Menu System          *" << endl;
        cout << "*****" << endl;
        cout << endl;
        int option;
        cout << "Select one of the following options number..." << endl;
        cout << "1. Enter Option 1" << endl;
        cout << "2. Enter Option 2" << endl;
        cout << "3. Enter Option 3" << endl;
        cout << "Your Option..";
        cin >> option;
        if (option == 1){
            cout << "You have Entered Option 1";
        }
        if (option == 2){
            cout << "You have Entered Option 2";
        }
        if (option == 3){
            cout << "You have Entered Option 3";
        }
    }
}

```

We can make a function
to just display the
header.

```

#include <iostream>
using namespace std;
main()
{
    cout << "*****" << endl;
    cout << "          Console Based Menu System          *" << endl;
    cout << "*****" << endl;
    cout << endl;
    int option;
    cout << "Select one of the following options number..." << endl;
    cout << "1. Enter Option 1" << endl;
    cout << "2. Enter Option 2" << endl;
    cout << "3. Enter Option 3" << endl;
    cout << "Your Option..";
    cin >> option;
    if (option == 1){
        cout << "You have Entered Option 1";
    }
    if (option == 2){
        cout << "You have Entered Option 2";
    }
    if (option == 3){
        cout << "You have Entered Option 3";
    }
}

```

We can make a function to display the menu and take the input from the user

| Solution

Lets see the solution with functions

```
void header()  
{  
    cout << "*****" << endl;  
    cout << "          Console Based Menu System          *" << endl;  
    cout << "*****" << endl;  
    cout << endl;  
}
```

```
int menu()  
{  
    int option;  
    cout << "Select one of the following options number..." << endl;  
    cout << "1. Enter Option 1" << endl;  
    cout << "2. Enter Option 2" << endl;  
    cout << "3. Enter Option 3" << endl;  
    cout << "Your Option..";  
    cin >> option;  
    return option;  
}
```

```
#include <iostream>
using namespace std;
void header();
int menu();
main()
{
    int option;
    header();
    option = menu();
    if (option == 1)
    {
        cout << "You have Entered Option 1";
    }
    if (option == 2)
    {
        cout << "You have Entered Option 2";
    }
    if (option == 3)
    {
        cout << "You have Entered Option 3";
    }
}
```

|| Benefits of Functions

Now, The code looks **organized** and **readable**.
We can easily understand the functionality of the code by looking at the **main function** now.

University Admission Management System

Let's move towards the more **realistic** application.



UAMS: Requirements

Make a University Admission Management System with the following options now.

```
*****  
*           University Admission Management System           *  
*****  
  
Select one of the following options number...  
1. Add First Student Record  
2. Add Second Student Record  
3. Calculate Aggregates  
4. Print Both Students Data  
5. Exit  
Your Option..
```


UAMS: Option 1

```
*****
*           University Admission Management System           *
*****

Select one of the following options number...
1. Add First Student Record
2. Add Second Student Record
3. Calculate Aggregates
4. Print Both Students Data
5. Exit
Your Option..1
Enter Student Name...Ali
Enter Obtained Marks in Matric..1000
Enter Obtained Marks in First Year..490
Enter Ecat Marks..280
Press any Key to Continue: █
```

UAMS: Option 2

```
*****
*           University Admission Management System           *
*****

Select one of the following options number...
1. Add First Student Record
2. Add Second Student Record
3. Calculate Aggregates
4. Print Both Students Data
5. Exit
Your Option..2
Enter Student Name...Ahmad
Enter Obtained Marks in Matric..1040
Enter Obtained Marks in First Year..500
Enter Ecat Marks..350
Press any Key to Continue:
```

UAMS: Option 3

```
*****
*           University Admission Management System           *
*****

Select one of the following options number...
1. Add First Student Record
2. Add Second Student Record
3. Calculate Aggregates
4. Print Both Students Data
5. Exit
Your Option..3
Press any Key to Continue:
```

UAMS: Option 4

```
*****
*           University Admission Management System           *
*****

Select one of the following options number...
1. Add First Student Record
2. Add Second Student Record
3. Calculate Aggregates
4. Print Both Students Data
5. Exit
Your Option..4
Following Students Exist in the System
Name      Matric   FYear   eCat    Merit
Ali        1000     490     280     84.9004
Ahmad     1040     500     350     91.921
Press any Key to Continue:
```

UAMS: Option 5

```
*****
*           University Admission Management System           *
*****

Select one of the following options number...
1. Add First Student Record
2. Add Second Student Record
3. Calculate Aggregates
4. Print Both Students Data
5. Exit
Your Option..5

G:\Programming Fundamentals (Fall 2022)\Week 5>
```

|| Solution

Let's first make the function of the header and menu.

Solution

```
void header()
{
    cout << "*****" << endl;
    cout << "          University Admission Management System          *" << endl;
    cout << "*****" << endl;
    cout << endl << endl;
}

int menu()
{
    int option;
    cout << "Select one of the following options number..." << endl;
    cout << "1. Add First Student Record" << endl;
    cout << "2. Add Second Student Record" << endl;
    cout << "3. Calculate Aggregates" << endl;
    cout << "4. Print Both Students Data" << endl;
    cout << "5. Exit" << endl;
    cout << "Your Option..";
    cin >> option;
    return option;
}
```

Main Function

```
#include <iostream>
using namespace std;
void header();
int menu();
int main()
{
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1) { }
        if (option == 2) { }
        if (option == 3) { }
        if (option == 4) { }
        if (option == 5) { }
    }
}
```


Main Function

We have to make the user allow to see the screen until he/she presses any key then clear the previous screen.

```
#include <iostream>
using namespace std;
void header();
int menu();
int main()
{
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1) { }
        if (option == 2) { }
        if (option == 3) { }
        if (option == 4) { }
        if (option == 5) { }
    }
}
```

Main Function

We have to make the user allow to see the screen until he/she presses any key then clear the previous screen.

```
#include <iostream>
using namespace std;
void header();
int menu();
int main()
{
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1) { }
        if (option == 2) { }
        if (option == 3) { }
        if (option == 4) { }
        if (option == 5) { }
        cout << "Press any Key to Continue: ";
        getch();
        system("cls");
    }
}
```

Main Function

getch() function stands for get character from the console. It will wait for the further execution until the user presses any keyboard key.

```
#include <iostream>
using namespace std;
void header();
int menu();
int main()
{
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1) { }
        if (option == 2) { }
        if (option == 3) { }
        if (option == 4) { }
        if (option == 5) { }
        cout << "Press any Key to Continue: ";
        getch();
        system("cls");
    }
}
```

Main Function

`getch()` function is defined in the `conio.h` library therefore, we have to include it before using it.

```
#include <iostream>
using namespace std;
void header();
int menu();
int main()
{
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1) { }
        if (option == 2) { }
        if (option == 3) { }
        if (option == 4) { }
        if (option == 5) { }
        cout << "Press any Key to Continue: ";
        getch();
        system("cls");
    }
}
```

Main Function

`getch()` function is defined in the `conio.h` library therefore, we have to include it before using it.

```
#include <iostream>
#include <conio.h>
using namespace std;
void header();
int menu();
int main()
{
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1) { }
        if (option == 2) { }
        if (option == 3) { }
        if (option == 4) { }
        if (option == 5) { }
        cout << "Press any Key to Continue: ";
        getch();
        system("cls");
    }
}
```

Solution: Option 1

```
if (option == 1)
{
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}
```

Solution: Option 2

```
if (option == 2)
{
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
```

Solution: Option 3

```
if (option == 3)
{
merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}
```


Solution: Option 4

```
if (option == 4)
{
cout << "Following Students Exist in the System" << endl;
cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" <<
endl;
cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" <<
merit1 << endl;
cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" <<
merit2 << endl;
}
```

|| Solution: Option 5

```
if (option == 5)
{
    return 0;
}
```

Do you see any of the code repeating in any of the options?

```
if (option == 1){
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}

if (option == 2){
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}

if (option == 3){
    merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
    merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}

if (option == 4){
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" << merit1 << endl;
    cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" << merit2 << endl;
}
```

Can you identify the code?

```
if (option == 1){
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}
if (option == 2){
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3){
    merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
    merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}
if (option == 4){
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" << merit1 << endl;
    cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" << merit2 << endl;
}
```

Here we are calculating the merit by using the same formula twice.

```
if (option == 1){
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}
if (option == 2){
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3){
    merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
    merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}
if (option == 4){
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" << merit1 << endl;
    cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" << merit2 << endl;
}
```

We can further improve the code readability by making a function that will take information of the student and calculate its aggregate

```
if (option == 1){
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}
if (option == 2){
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3){
    merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
    merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}
if (option == 4){
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" << merit1 << endl;
    cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" << merit2 << endl;
}
```

Here we are printing
the data twice.

```
if (option == 1){
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}
if (option == 2){
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3){
    merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
    merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}
if (option == 4){
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" << merit1 << endl;
    cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" << merit2 << endl;
}
```

We can further improve the code readability by making a function that will take information of the student and print it on console in specific format.

```
if (option == 1){
    cout << "Enter Student Name...";
    cin >> sName1;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks1;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks1;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks1;
}
if (option == 2){
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3){
    merit1 = ((0.25*(matricMarks1/1050)) + (0.45*(firstYearMarks1/550)) + (0.3*(eCatMarks1/400))) * 100;
    merit2 = ((0.25*(matricMarks2/1050)) + (0.45*(firstYearMarks2/550)) + (0.3*(eCatMarks2/400))) * 100;
}
if (option == 4){
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    cout << sName1 << "\t" << matricMarks1 << "\t" << firstYearMarks1 << "\t" << eCatMarks1 << "\t" << merit1 << endl;
    cout << sName2 << "\t" << matricMarks2 << "\t" << firstYearMarks2 << "\t" << eCatMarks2 << "\t" << merit2 << endl;
}
```


|| Solution: **With Functions**

Let's further make the functions of aggregate and displayStudentData.

Solution: Previous Functions

```
void header()
{
    cout << "*****" << endl;
    cout << "          University Admission Management System          *" << endl;
    cout << "*****" << endl;
    cout << endl << endl;
}

int menu()
{
    int option;
    cout << "Select one of the following options number..." << endl;
    cout << "1. Add First Student Record" << endl;
    cout << "2. Add Second Student Record" << endl;
    cout << "3. Calculate Aggregates" << endl;
    cout << "4. Print Both Students Data" << endl;
    cout << "5. Exit" << endl;
    cout << "Your Option..";
    cin >> option;
    return option;
}
```

Solution: New Functions

```
float calculateAggregate(float matricMarks, float firstYearMarks, float eCatMarks)
{
    float aggregate;
    aggregate = ((0.25 * (matricMarks / 1050)) + (0.45 * (firstYearMarks / 550)) + (0.3 * (eCatMarks / 400))) *
100;
    return aggregate;
}

void printStudentData(string sName, float matricMarks, float firstYearMarks, float eCatMarks, float merit)
{
    cout << sName << "\t" << matricMarks << "\t" << firstYearMarks << "\t" << eCatMarks << "\t" << merit <<
endl;
}
```

```
#include <iostream>
#include <conio.h>
using namespace std;

void header();
int menu();
float calculateAggregate(float matricMarks, float firstYearMarks, float eCatMarks);
void printStudentData(string sName, float matricMarks, float firstYearMarks, float eCatMarks, float merit);
int main(){
    int option;
    string sName1 = "", sName2 = "";
    float matricMarks1 = 0, matricMarks2 = 0;
    float firstYearMarks1 = 0, firstYearMarks2 = 0;
    float eCatMarks1 = 0, eCatMarks2 = 0;
    float merit1 = 0, merit2 = 0;
    while (true)
    {
        header();
        int option = menu();
        if (option == 1)
        {
            cout << "Enter Student Name...";
            cin >> sName1;
            cout << "Enter Obtained Marks in Matric..";
            cin >> matricMarks1;
            cout << "Enter Obtained Marks in First Year..";
            cin >> firstYearMarks1;
            cout << "Enter Ecat Marks..";
            cin >> eCatMarks1;
        }
    }
}
```

```
if (option == 2)
{
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3)
{
    merit1 = calculateAggregate(matricMarks1, firstYearMarks1, eCatMarks1);
    merit2 = calculateAggregate(matricMarks2, firstYearMarks2, eCatMarks2);
}
if (option == 4)
{
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    printStudentData(sName1, matricMarks1, firstYearMarks1, eCatMarks1, merit1);
    printStudentData(sName2, matricMarks2, firstYearMarks2, eCatMarks2, merit2);
}
if (option == 5)
{
    return 0;
}
cout << "Press any Key to Continue: ";
getch();
system("cls");
}
```

We can further improve the code by making separate two functions for these functionalities.

```
if (option == 2)
{
    cout << "Enter Student Name...";
    cin >> sName2;
    cout << "Enter Obtained Marks in Matric..";
    cin >> matricMarks2;
    cout << "Enter Obtained Marks in First Year..";
    cin >> firstYearMarks2;
    cout << "Enter Ecat Marks..";
    cin >> eCatMarks2;
}
if (option == 3)
{
    merit1 = calculateAggregate(matricMarks1, firstYearMarks1, eCatMarks1);
    merit2 = calculateAggregate(matricMarks2, firstYearMarks2, eCatMarks2);
}
if (option == 4)
{
    cout << "Following Students Exist in the System" << endl;
    cout << "Name" << "\t" << "Matric" << "\t" << "FYear" << "\t" << "eCat" << "\t" << "Merit" << endl;
    printStudentData(sName1, matricMarks1, firstYearMarks1, eCatMarks1, merit1);
    printStudentData(sName2, matricMarks2, firstYearMarks2, eCatMarks2, merit2);
}
if (option == 5)
{
    return 0;
}
cout << "Press any Key to Continue: ";
getch();
system("cls");
}
```

Learning Outcome

Categorize the code into **meaningful functions** to make the code more **modular, readable, structured, and reusable.**

