18F-0336

Section: C

Muhammad Ahmer Zaman

Question1:

You are required to create an array of struct type students. In which you are required to store the

gender of the student using the ENUM. Also, you need to use pointers to store student

information in the heap. Student should have following information

• First Name

• Last Name

• Roll No

• Gender

Code:

#include <iostream>

#include <string>

using namespace std;

enum Gender {

Male,

Female,

};

struct Student {

int Roll\_Number;

string First\_Name;

string Last\_Name;

Gender Student\_Gender;

};

void main()

{

int Gen;

int max\_studs;

cout << "Enter Number of Students You Want to Enter!" << endl;

do

{

cin >> max\_studs;

cout << endl << endl;

} while ((max\_studs <= 1) && (max\_studs >= 1000));

Student\*S = new Student[max\_studs];

for (int i = 0; i < max\_studs; i++)

{

cout << "Student Number " << i + 1 << endl;

cout << "First Name:";

cin >> S[i].First\_Name;

cout << "Last Name:";

cin >> S[i].Last\_Name;

do

{

cout << "Roll Number# 18F-XXXX" << endl << "18F-";

cin >> S[i].Roll\_Number;

} while ((S[i].Roll\_Number < 0)||(S[i].Roll\_Number > 9999));

do

{

cout << "Gender: (0 for male or 1 for female)";

cin >> Gen;

} while ((Gen != 0) && (Gen != 1));

cout << endl << endl;

}

for (int i = 0; i < max\_studs; i++)

{

if (Gen == 0)

{

S[i].Student\_Gender = Male;

}

else

{

S[i].Student\_Gender = Female;

}

}

for (int i = 0; i < max\_studs; i++)

{

cout << "Student Number " << i + 1 << endl;

cout << "First Name:";

cout << S[i].First\_Name << endl;

cout << "Last Name:";

cout << S[i].Last\_Name << endl;

if (S[i].Roll\_Number < 1000)

{

cout << "Roll Number# 18F-0";

cout << S[i].Roll\_Number << endl;

}

else

{

cout << "Roll Number# 18F- ";

cout << S[i].Roll\_Number << endl;

}

cout << "Gender:";

if (S[i].Student\_Gender = Male)

{

cout << "Male" << endl;

}

else

{

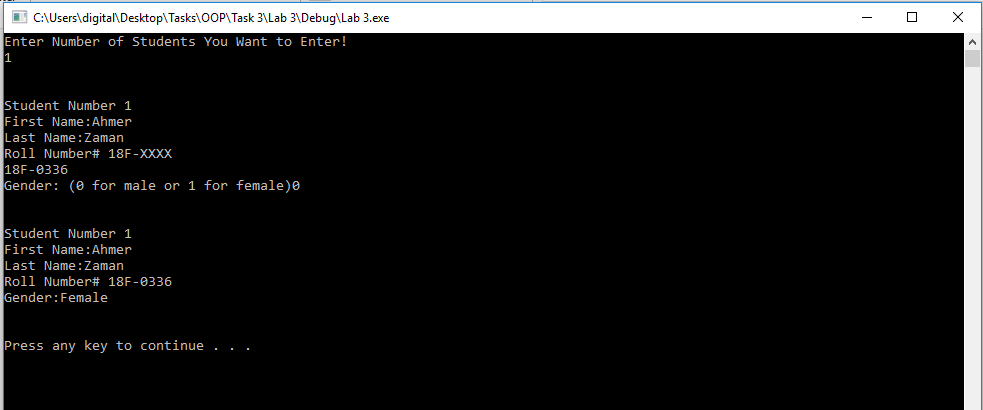
cout << "Female" << endl;

}

cout << endl << endl;

}

system("pause");

}

Question2:

You are developing a banking application and your task is related to the credit card information

of the customers. And there are 3 type of cards Gold, Silver and Platinum and your task is to

assign some specific amount of discount based on the customer card type and also based on the

Day of the week. But when you try to code it becomes very difficult for you to remember what

represents Gold or what represents Wednesday. So you take a better approach to declare

Enumeration for it.

Let’s assume the discount values:

Platinum = 10%

Gold = 7%

Silver = 4%

And the discount on the day of the week is based on this function Discount(Day) =

EnumValue(D)\*2.

And you start your week with Monday.

Now your task is to your customers comes goes for some shopping and pays with his/her credit

card. Now you have to Input the Amount of user spent and what the Bank is going to charge him

based on the discounts calculated above.

Code:

#include <iostream>

#include <string>

using namespace std;

enum Card {

Platinum,

Gold,

Silver,

};

enum disc{

p = 10,

g = 7,

s = 4,

};

enum Day {

Monday,

Tuesday,

Wednessday,

Thursday,

Friday,

Saturday,

Sunday,

};

int Discount(int D,int C);

void main()

{

int discount = 0,Amount=0;

int A, type;

do

{

cout << "Enter Card Type: " << endl;

cout << "0 for Platinum/1 for Gold/2 for Silver" << endl;

cin >> type;

} while ((type <-10) || (type >3));

do

{

cout << "Enter Day of Week 0.Monday,1.Tuesday,....7,Sunday" << endl;

cin >> A;

} while ((A < -1) || (A > 8));

cout << "Enter Total Amount: " << endl;

cin >> Amount;

cout << endl << endl;

if (type == Platinum)

{

discount = (Amount\*Discount(A,p));

cout << "Discounted Amount: " <<Amount- discount<< endl;

}

else if (type == Gold)

{

discount =(Amount\*Discount(A,g));

cout << "Discounted Amount: " << Amount - discount << endl;

}

else

{

discount =(Amount\*Discount(A,s));

cout << "After Discounted Amount is: " <<Amount - discount << endl;

}

system("pause>0");

}

int Discount(int D,int C)

{

if (D == Monday)

{

return (D\*C)/100;

}

else if (D==Tuesday)

{

return (D\*C)/100;

}

else if (D == Wednessday)

{

return (D\*C)/100;

}

else if (D == Thursday)

{

return (D\*C)/100;

}

else if (D == Friday)

{

return (D\*C)/100;

}

else if (D == Saturday)

{

return (D\*C)/100;

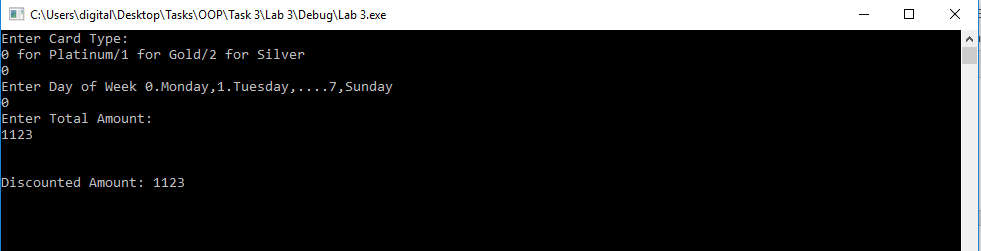
}

else

{

return (D\*C)/100;

}

}

Question3:

You all might be familiar with TMP (if not ask your seniors). Let’s assume for now it is a

restaurant and want to computerize thier billing system. Now they have hired you to make a

menu driven program that asks for different menu items and calculates bill accordingly.

As you have recently learned about the Structs and ENUMS so you think of a better approach of

making this system. First of all, you create an Order Struct to store the Menu items their quantity

and amount. For the menu items you might be using ENUM as for the long list of Menu you

don’t want to memorize each item id. Then you pass this Order struct to a function Calculate Bill

and it returns you the total Bill amount that you display on the screen.

Code:

#include <iostream>

using namespace std;

enum Dish {

Rice,

Gravy,

Sweet

};

struct Order {

int Dish\_No;

int Servings;

};

int Calculate\_Bill(Order S);

void main()

{

char Again;

int size = 0, Total\_Bill=0;

do

{

cout << "Enter Max number of Customers you want to Enter: ";

cin >> size;

} while ((size < 0)||(size>101));

Order \* S = new Order[size];

int \* Bill = new int [size];

for (int i = 0, Total\_Bill=0; i < size; i++)

{

cout << "Order Number" << i + 1 << endl << endl;

do {

do

{

cout << "Enter Dish Number: 0 for Rice, 1 for Gravy, 2 For Sweet" << endl;

cin >> S[i].Dish\_No;

} while ((S[i].Dish\_No < 0) || (S[i].Dish\_No > 2) );

do

{

cout << "Enter Servings" << endl;

cin >> S[i].Servings;

} while (S[i].Servings < 0);

Total\_Bill = Total\_Bill + Calculate\_Bill(S[i]);

cout << "Do You Want to Order More? Y/N" << endl;

cin >> Again;

} while (Again == 'Y' || Again == 'y');

cout << "Total Bill of Customer No" << i + 1 << " is Rs." << Total\_Bill << endl << endl;

Total\_Bill = Bill[i];

}

system("pause>0");

}

int Calculate\_Bill(Order S)

{

if (S.Dish\_No == Rice)

{

return 100 \*S.Servings;

}

else if (S.Dish\_No == Gravy)

{

return 70 \* S.Servings;

}

else

{

return 50 \* S.Servings;

}

}

