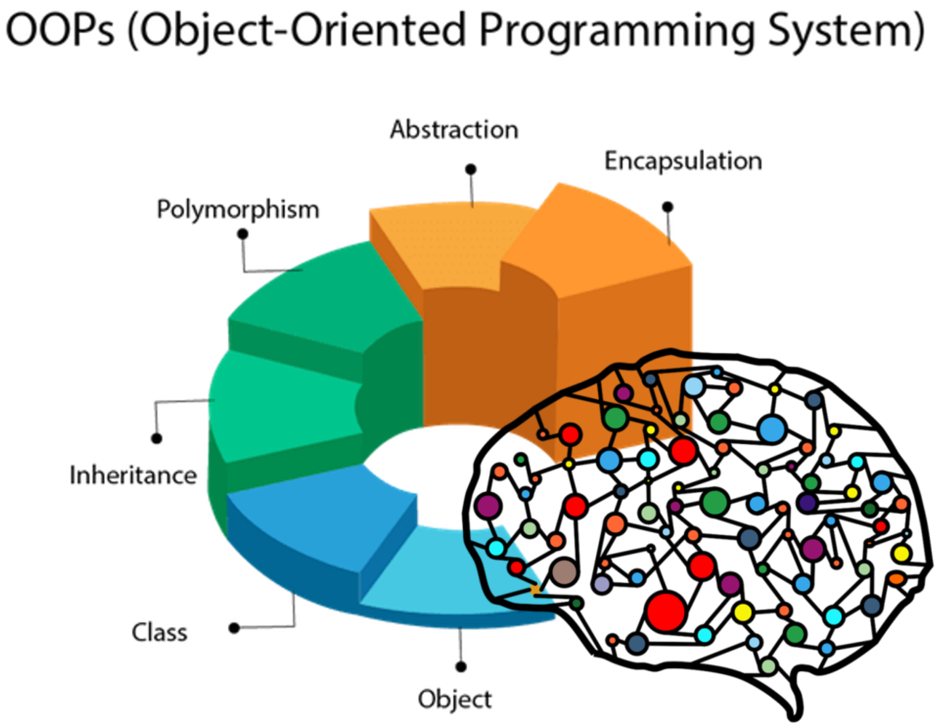
ASSIGNMENT -3

PERSISTENCE OF PROJECT



Presented by:
Abhisar Sen
Priyanshi Mulgaonkar
Vidushi Yadav

PROBLEM STATEMENT

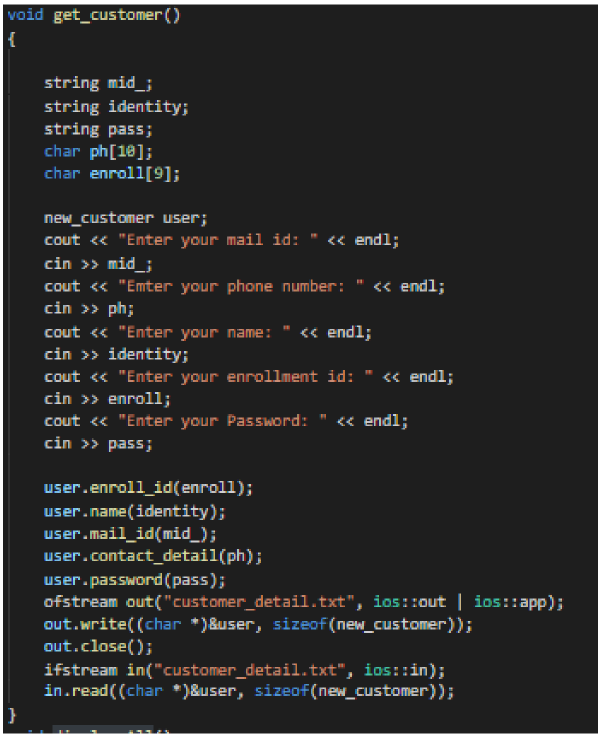
In this assignment, the students are required to build File Storage Option for the corresponding Project Assignment and store all the data generated into these files successfully.
The students will also be required to demonstrate the run-time / execution screen shots of their projects. Same projects as been given in the first two assignments would be used for Assignment -3
The projects as assigned to students should be used for this purpose..
So we have continued the **canteen management system** in this assignment..

CODE HIGHLIGHT

Some modification are done in the code ahead of the Task 2 those functions are listed here -

VOID GET_CUSTOMER()

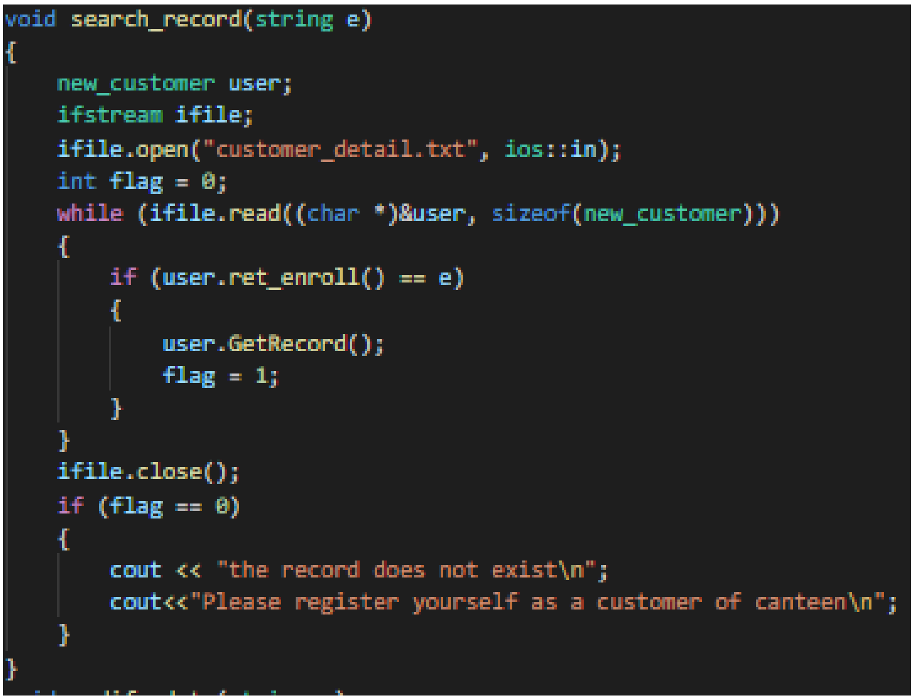
For permanently adding the data of customer in files.



```cpp
void get_customer()
{

    string mid_;
    string identity;
    string pass;
    char ph[10];
    char enroll[9];

    new_customer user;
    cout << "Enter your mail id: " << endl;
    cin >> mid_;
    cout << "Emter your phone number: " << endl;
    cin >> ph;
    cout << "Enter your name: " << endl;
    cin >> identity;
    cout << "Enter your enrollment id: " << endl;
    cin >> enroll;
    cout << "Enter your Password: " << endl;
    cin >> pass;

    user.enroll_id(enroll);
    user.name(identity);
    user.mail_id(mid_);
    user.contact_detail(ph);
    user.password(pass);
    ofstream out("customer_detail.txt", ios::out | ios::app);
    out.write((char *)&user, sizeof(new_customer));
    out.close();
    ifstream in("customer_detail.txt", ios::in);
    in.read((char *)&user, sizeof(new_customer));
}
```

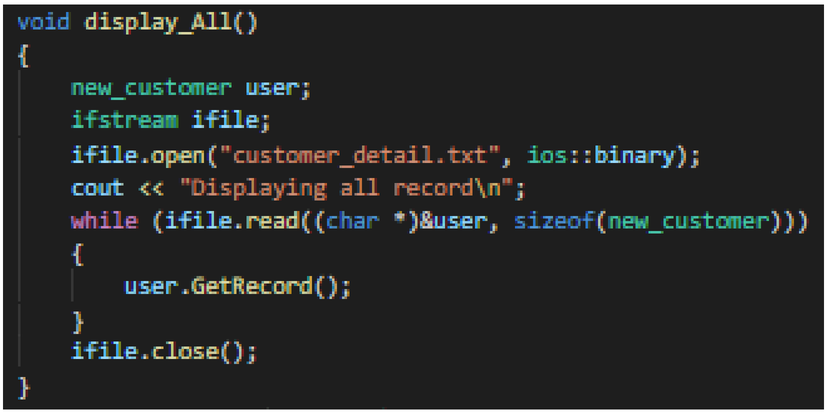
VOID SEARCH_RECORD()

For searching a particular record of customer



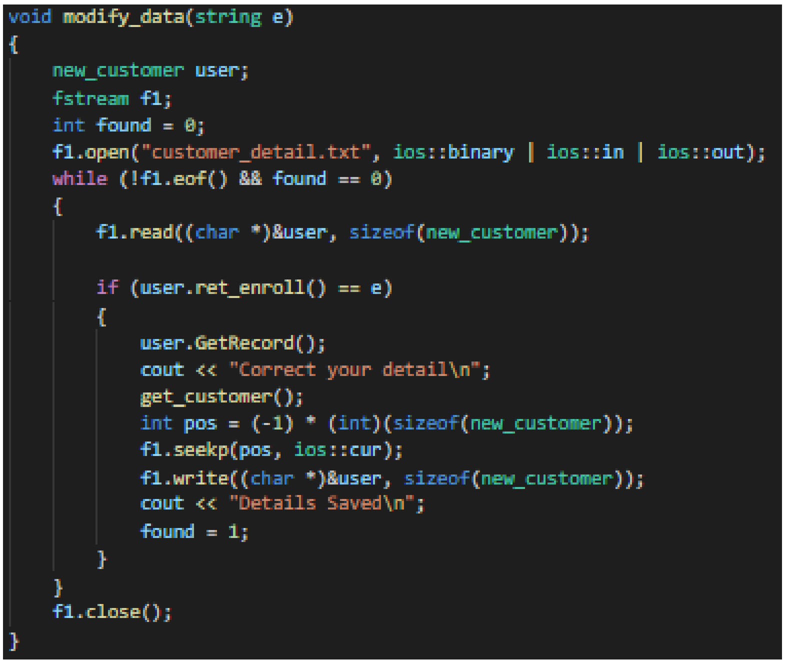
```cpp
void search_record(string e)
{
    new_customer user;
    ifstream ifile;
    ifile.open("customer_detail.txt", ios::in);
    int flag = 0;
    while (ifile.read((char *)&user, sizeof(new_customer)))
    {
        if (user.ret_enroll() == e)
        {
            user.GetRecord();
            flag = 1;
        }
    }
    ifile.close();
    if (flag == 0)
    {
        cout << "the record does not exist\n";
        cout<<"Please register yourself as a customer of canteen\n";
    }
}
```

VOID DISPLAY_ALL()



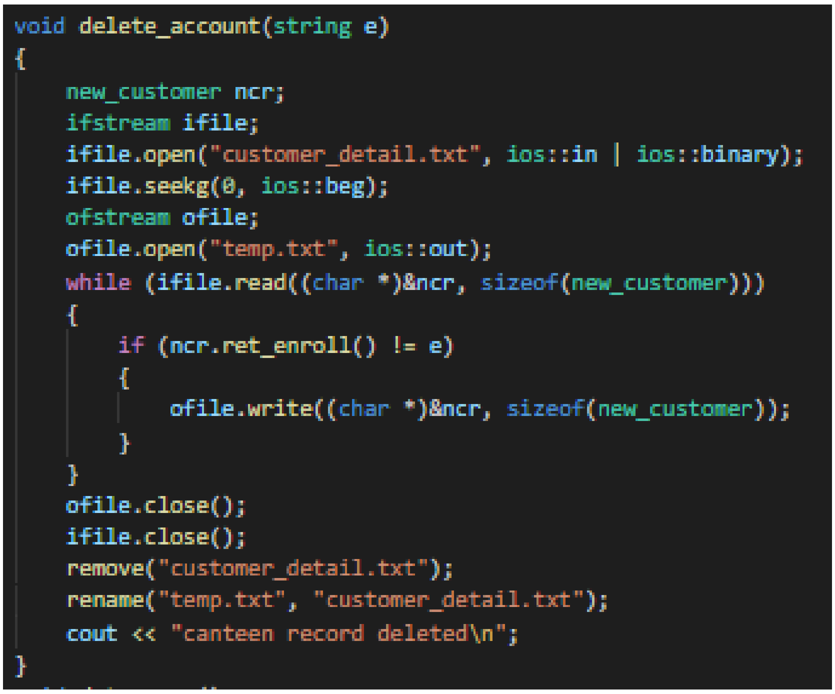
```
void display_All()
{
    new_customer user;
    ifstream ifile;
    ifile.open("customer_detail.txt", ios::binary);
    cout << "Displaying all record\n";
    while (ifile.read((char *)&user, sizeof(new_customer)))
    {
        user.GetRecord();
    }
    ifile.close();
}
```

VOID MODIFY_DETAIL(STRING)



```
void modify_data(string e)
{
    new_customer user;
    fstream f1;
    int found = 0;
    f1.open("customer_detail.txt", ios::binary | ios::in | ios::out);
    while (!f1.eof() && found == 0)
    {
        f1.read((char *)&user, sizeof(new_customer));

        if (user.ret_enroll() == e)
        {
            user.GetRecord();
            cout << "Correct your detail\n";
            get_customer();
            int pos = (-1) * (int)(sizeof(new_customer));
            f1.seekp(pos, ios::cur);
            f1.write((char *)&user, sizeof(new_customer));
            cout << "Details Saved\n";
            found = 1;
        }
    }
    f1.close();
}
```

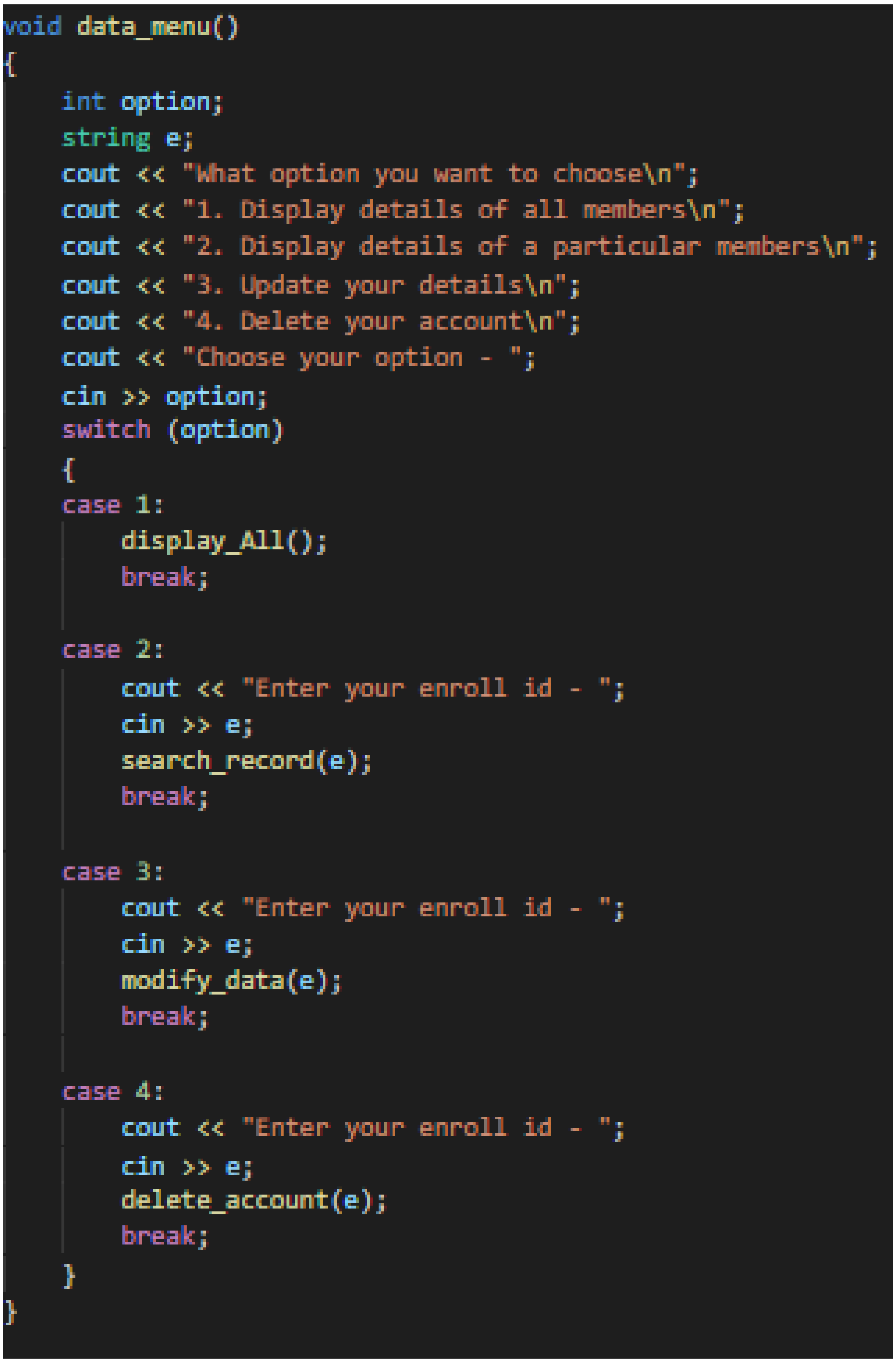
VOID DELETE ACCOUNT(STRING)



```
void delete_account(string e)
{
    new_customer ncr;
    ifstream ifile;
    ifile.open("customer_detail.txt", ios::in | ios::binary);
    ifile.seekg(0, ios::beg);
    ofstream ofile;
    ofile.open("temp.txt", ios::out);
    while (ifile.read((char *)&ncr, sizeof(new_customer)))
    {
        if (ncr.ret_enroll() != e)
        {
            ofile.write((char *)&ncr, sizeof(new_customer));
        }
    }
    ofile.close();
    ifile.close();
    remove("customer_detail.txt");
    rename("temp.txt", "customer_detail.txt");
    cout << "canteen record deleted\n";
}
```

HOW THESE FUNCTIONS WILL BE CALEED-

void data_menu()



```
void data_menu()
{
    int option;
    string e;
    cout << "What option you want to choose\n";
    cout << "1. Display details of all members\n";
    cout << "2. Display details of a particular members\n";
    cout << "3. Update your details\n";
    cout << "4. Delete your account\n";
    cout << "Choose your option - ";
    cin >> option;
    switch (option)
    {
    case 1:
        display_All();
        break;

    case 2:
        cout << "Enter your enroll id - ";
        cin >> e;
        search_record(e);
        break;

    case 3:
        cout << "Enter your enroll id - ";
        cin >> e;
        modify_data(e);
        break;

    case 4:
        cout << "Enter your enroll id - ";
        cin >> e;
        delete_account(e);
        break;
    }
}
```

DETAILED CODE -

```
#include <iostream>
#include <cstring>
#include <fstream>
#include <stdio.h>
using namespace std;
int wish;
void modify_data(int);
void dash(int no)
{
  for (int i = 0; i < no; i++)
  {
    cout << "-";
  }
  cout << endl;
}
class new_customer
{
  char id[9];
  char ph_no[10];
  string customer;
  string mail;
  string security;

public:
  void enroll_id(char *p)
  {
    strcpy(id, p);
  }
  void contact_detail(char *p)
  {
    strcpy(ph_no, p);
  }
  void name(string cust)
  {
    customer = cust;
  }
  void mail_id(string mailid)
  {
    mail = mailid;
  }
```

```
void password(string pass)
 {
 security = pass;
 }
 void GetRecord()
 {
 cout << "Student Enroll ID\t" << id << endl;
 cout << "Student Contact Nuber\t" << ph_no << endl;
 cout << "Student name\t" << customer << endl;
 cout << "Student mail id\t" << mail << endl;
 // cout<<"\n\n";
 }
 string ret_enroll()
 {
 return id;
 }
 string ret_pass()
 {
 return security;
 }
};

class customer_info
{
  char id[9];
  char password[10];

public:
  void enroll_id(char *p)
  {
    strcpy(id, p);
  }

  void pass(char *p)
  {
    strcpy(password, p);
  }
};

class Food_detail
{
public:
  void breakfast()
  {
    dash(55);
    cout << "S.no.\t|\tItem\t|\tPrice/Platen(in Rs.)" << endl;
    dash(55);
    cout << "1.\t|\tTea\t|\t10" << endl;
    cout << "2.\t|\tCofee\t|\t10" << endl;
    cout << "3.\t|\tPoha\t|\t30" << endl;
    cout << "\n\n";
  }
```

```
void Lunch()
{
  cout << "Following will be there in a plate for lunch" << endl;
  dash(40);
  cout << "S.no.\t|\tItem" << endl;
  dash(40);
  cout << "1.\t|\tMatar Paneer" << endl;
  cout << "2.\t|\tSev ki Sabji" << endl;
  cout << "3.\t|\tTawa Roti" << endl;
  cout << "4.\t|\tGulaab Jamun" << endl;
  cout << "5.\t|\tAchaar/Namkeen" << endl;
  dash(40);
  cout << "Price of per plate is 70Rs." << endl;
  cout << "\n\n";
}
void snacks()
{
  cout << "Following will be there in snacks for today" << endl;
  dash(55);
  cout << "S.no.\t|\tItem\t|\tPrice/palte(in Rs.)" << endl;
  dash(55);
  cout << "1.\t|\tBhel\t|\t25" << endl;
  cout << "2.\t|\tDhokla\t|\t30" << endl;
  cout << "3.\t|\tMaggie\t|\t25" << endl;
  cout << "4.\t|\tTea\t|\t10" << endl;
  cout << "5.\t|\tCoffee\t|\t10" << endl;
  cout << "6.\t|\tPakode\t|\t30" << endl;
  dash(55);
  cout << "\n\n";
}
void Dinner()
{
  cout << "Following will be there in one plate for dinner for tonight" << endl;
  dash(40);
  cout << "S.no.\t|\tItem\t|\t)" << endl;
  dash(40);
  cout << "1.\t|\tPalak Paneer" << endl;
  cout << "2.\t|\tKadhi" << endl;
  cout << "3.\t|\tAchaar/Namkeen" << endl;
  cout << "4.\t|\tTandoori Roti" << endl;
  cout << "5.\t|\tHalwa" << endl;
  dash(40);
  cout << "\n\n";
}
void Special()
{
  cout << "Following will be as an special item" << endl;
  dash(55);
  cout << "S.no.\t|\tItem\t|\tPrice/item(in Rs.))" << endl;
  dash(55);
  cout << "1.\t|\tIce-Cream\t|\t15" << endl;
  cout << "2.\t|\tChocolates\t|\t10" << endl;
  cout << "3.\t|\tPaani Puri\t|\t10" << endl;
  dash(55);
```

```
   cout << "\n\n";
 }
};
class Menu_Detail : public Food_detail
{
public:
 void m_detail()
 {
  int choice;
  cout << "\n You have to choose the following..";
  cout << "\n1.Purchase Detail";
  cout << "\n2.Show Menu";
  cout << "\n3.Rules and Regulations";
  cout << "\n4.Food Suggestion";
  cout << "\nEnter your choice: ";
  cin >> choice;
  if (choice == 2)
  {
   cout << "Which menu do you want to know"
     << "\n 1.Breakfast \n 2.Lunch \n 3.Snacks \n 4.Dinner \n 5.Special Food" << endl;
   cout << "Enter your choice:";
   cin >> wish;
   if (wish == 1)
   {
    breakfast();
   }
   else if (wish == 2)
   {
    // Lunch
    Lunch();
   }
   else if (wish == 3)
   {
    // Snacks
    snacks();
   }
   else if (wish == 4)
   {
    // Dinner
    Dinner();
   }
   else if (wish == 5)
   {
    // Special food
    Special();
   }
  }
  else if (choice == 3)
  {
   cout << "\t\t\t\t\t\t\tThe Rules and Regulation of Avantika Canteen" << endl;
   cout << "All used cutlery, plates and trays MUST be returned to the assigned stacking areas.
Please do not leave them on the tables." << endl;
```

```
 cout << "Cutlery, plates and any other canteen utensils are not to be taken to classes, library or
away from the canteen area." << endl;
 cout << "A refundable fee of ksh.100 will be charged for every glass picked from the canteen
serving points. This is refunded immediately you return the glass back." << endl;
 cout << "Practice good table manners. Keep all food on the tray and off tables and floors." << endl;
 }
else if (choice == 4)
    {
      string suggestion;
      cout << "Enter your kind suggestion here: ";
      cin >> suggestion;
      cout << "\nYour suggestion is very helpful for us. Thank You!" << endl;
    }
  }
};
class purchase : public Menu_Detail
{
public:
  purchase(int a)
  {
    cout << "\tHope you are enjoying your food at \n\t\tAVANTIKA UNIVERSITY" << endl;
  }
  purchase()
  {
  }

  void invoice()
  {
    cout << "\n\n";
    int option, total = 0;
    int a = wish;
    switch (a)
    {
    case 1:
      cout << "\tBreakfast Purchased\n";
      cout << "what thing do you want to order?\nEnter order S.No.\n";
      cout << "\nKindly Press 0 for completing oreder\n";

      cout << "your purchase detail is given below " << endl;
      dash(45);
      cout << "Item\t|\tQty.\t|\tRate" << endl;
      dash(45);
      while (option > 0)
      {
        cin >> option;
        if (option == 1 | option == 2)
        {
          total = total + 10;
          if (option == 1)
          {
            cout << "tea\t|\t1\t|\t10\n";
          }
     else
    {
    cout << "Coffee\t|\t1\t|\t10\n";
    }
    }
    else if (option == 3)
    {
    total = total + 30;
    cout << "poha\t|\t1\t|\t30\n";
    }
    }

         cout << "your total payable amount is - " << total << endl;
         break;
       case 2:
         cout << "\tLunch Purchased\n";
         int n;
         cout << "How many plates do you want to order?\n";
         cin >> n;
         total = n * 70;

         cout << "your total payable amount is - " << total << endl;
         break;
       case 3:
         cout << "\tSnacks Purchased\n";
         cout << "what thing do you want to order?\nEnter order S.No.\n";
         cout << "\nKindly Press 0 for completing order\n";

         cout << "your purchase detail is given below " << endl;
         dash(45);
         cout << "Item\t|\tQty.\t|\tRate" << endl;
         dash(45);

         while (option > 0)
         {
           cin >> option;
           if (option == 1 | option == 3)
           {
             if (option == 1)
             {
               cout << "Bhel\t|\t1\t|\t25\n";
             }
             else if (option == 3)
             {
               cout << "Maggie\t|\t1\t|\t20\n";
             }
             total = total + 25;
           }
           else if (option == 2 | option == 6)
           {
             if (option == 2)
             {
               cout << "Dhokla\t|\t1\t|\t25\n";
             }
     else if (option == 6)
     {
     cout << "Pakoda\t|\t1\t|\t20\n";
     }
     total = total + 30;
     }
    else if (option == 4 | option == 5)
        {
         if (option == 4)
         {
          cout << "Tea\t|\t1\t|\t25\n";
         }
         else if (option == 5)
         {
          cout << "Coffee\t|\t1\t|\t20\n";
         }
         total = total + 10;
        }
       }
       cout << "your total payable amount is - " << total << endl;
       break;
      case 4:
       cout << "\tDinner Purchased\n";
       cout << "How many plates do you want to order?\n";
       cin >> n;
       total = n * 70;
       cout << "your total payable amount is - " << total << endl;
       break;

      default:
       cout << "Supper Purchased\n";
       cout << "what thing do you want to order?\nEnter order S.No.\n";
       cout << "\nKindly Press 0 for completing oreder\n";

       cout << "your purchase detail is given below " << endl;
       dash(45);
       cout << "Item\t|\tQty.\t|\tRate" << endl;
       dash(45);

       while (option > 0)
       {
        cin >> option;
        if (option == 1)
        {
         if (option == 1)
         {
          cout << "Ice-Cream\t|\t1\t|\t25\n";
         }
         total = total + 15;
        }
```

```
else if (option == 2 | option == 3)
    {
     if (option == 2)
     {
      cout << "Chocolate\t|\t1\t|\t25\n";
     }
     else if (option == 2)
     {
      cout << "Paani-Puri\t|\t1\t|\t25\n";
     }
     total = total + 10;
    }
   }
   cout << "your total payable amount is - " << total << endl;
   break;
  }
 }
};
void get_customer()
{

 string mid_;
 string identity;
 string pass;
 char ph[10];
 char enroll[9];

 new_customer user;
 cout << "Enter your mail id: " << endl;
 cin >> mid_;
 cout << "Emter your phone number: " << endl;
 cin >> ph;
 cout << "Enter your name: " << endl;
 cin >> identity;
 cout << "Enter your enrollment id: " << endl;
 cin >> enroll;
 cout << "Enter your Password: " << endl;
 cin >> pass;

 user.enroll_id(enroll);
 user.name(identity);
 user.mail_id(mid_);
 user.contact_detail(ph);
 user.password(pass);
 ofstream out("customer_detail.txt", ios::out | ios::app);
 out.write((char *)&user, sizeof(new_customer));
 out.close();
 ifstream in("customer_detail.txt", ios::in);
 in.read((char *)&user, sizeof(new_customer));
}
```

```
void display_All()
{
  new_customer user;
  ifstream ifile;
  ifile.open("customer_detail.txt", ios::binary);
  cout << "Displaying all record\n";
  while (ifile.read((char *)&user, sizeof(new_customer)))
  {
    user.GetRecord();
  }
  ifile.close();
}
void search_record(string e)
{
  new_customer user;
  ifstream ifile;
  ifile.open("customer_detail.txt", ios::in);
  int flag = 0;
  while (ifile.read((char *)&user, sizeof(new_customer)))
  {
    if (user.ret_enroll() == e)
    {
      user.GetRecord();
      flag = 1;
    }
  }
  ifile.close();
  if (flag == 0)
  {
    cout << "the record does not exist\n";
    cout<<"Please register yourself as a customer of canteen\n";
  }
}
void modify_data(string e)
{
  new_customer user;
  fstream f1;
  int found = 0;
  f1.open("customer_detail.txt", ios::binary | ios::in | ios::out);
  while (!f1.eof() && found == 0)
  {
    f1.read((char *)&user, sizeof(new_customer));

    if (user.ret_enroll() == e)
    {
      user.GetRecord();
      cout << "Correct your detail\n";
      get_customer();
      int pos = (-1) * (int)(sizeof(new_customer));
      f1.seekp(pos, ios::cur);
      f1.write((char *)&user, sizeof(new_customer));
      cout << "Details Saved\n";
```

```
 found = 1;
 }
 }
 f1.close();
}
void delete_account(string e)
{
  new_customer user;
  ifstream ifile;
  ifile.open("customer_detail.txt", ios::in | ios::binary);
  ifile.seekg(0, ios::beg);
  ofstream ofile;
  ofile.open("temp.txt", ios::out);
  while (ifile.read((char *)&user, sizeof(new_customer)))
  {
    if (user.ret_enroll() != e)
    {
      ofile.write((char *)&user, sizeof(new_customer));
    }
  }
  ofile.close();
  ifile.close();
  remove("customer_detail.txt");
  rename("temp.txt", "customer_detail.txt");
  cout << "canteen record deleted\n";
}
void data_menu()
{
  int option;
  string e;
  cout << "What option you want to choose\n";
  cout << "1. Display details of all members\n";
  cout << "2. Display details of a particular members\n";
  cout << "3. Update your details\n";
  cout << "4. Delete your account\n";
  cout << "Choose your option - ";
  cin >> option;
  switch (option)
  {
  case 1:
    display_All();
    break;

  case 2:
    cout << "Enter your enroll id - ";
    cin >> e;
    search_record(e);
    break;

  case 3:
    cout << "Enter your enroll id - ";
    cin >> e;
    modify_data(e);
    break;
```

```
e case 4:
 cout << "Enter your enroll id - ";
 cin >> e;
 delete_account(e);
 break;
 }
}
void existing_member()
{
  char enrollid[9];
  char password[10];
  customer_info user1;
  cout << "Enter your id: " << endl;
  cin >> enrollid;
  cout << "Enter your password " << endl;
  cin >> password;
  user1.enroll_id(enrollid);
  user1.pass(password);
  Menu_Detail md;
  md.m_detail();

  ofstream out("menu_detail.txt", ios::out);
  out.write((char *)&md, sizeof(Menu_Detail));
  out.close();
  ifstream in("menu_detail.txt", ios::in);
  in.read((char *)&md, sizeof(Menu_Detail));

  purchase pur(5);
  pur.invoice();

  ofstream("Purchase_detail.txt", ios::out);
  out.write((char *)&pur, sizeof(purchase));
  out.close();
  ifstream("purchase_detail.txt", ios::in);
  in.read((char *)&pur, sizeof(purchase));
}
main()
{
  cout << "\t\t\t\t\t\t\t Welcome To Avantika Canteen" << endl;
  cout << "\n Choose your option" << endl;
  cout << "\n1.Existing Member";
  cout << "\n2.New Member";
  cout << "\n3.Option";
  int choice;
  cout << "\nChoose 1/2/3 - ";
  cin >> choice;
  switch (choice)
  {
  case 1:
  {
    existing_member();
    break;
  }
 case 2:
 {
 get_customer();
 break;
 }
 case 3:
 {
 data_menu();
 break;
 }
 }
}
```

CONCLUSION

This project gave an advance learning of Object Oriented Programing. It made us able to learn about how to store the user data permanently in the hard drive of system, as in the project we stored the customer detail, his purchase history along with his order so that it an get used whenever it is required to need. In this task, we used one function to run the four different functions like modification, insertion, deletion and display of user data from the file stored in the hard drive to the console.