Task 1:

**Interface**:

class person

{

protected:

string first\_name;

string last\_name;

int age;

public:

person() {};

~person() {};

};

class student:public virtual person

{

protected:

float cgpa;

public:

student() {};

~student() {};

};

class faculty:public virtual person

{

private:

int no\_of\_courses;

int tel\_no;

public:

faculty() {};

~faculty() {};

void set(string a, string b, int ag, int ex);

void print();

};

class undergraduate :public virtual student

{

private:

string fyp\_name;

public:

undergraduate() {};

~undergraduate() {};

void set(string a, string b, string c, float g, int ag);

void print();

};

class graduate :public virtual student

{

private:

string thesis;

public:

graduate() {};

~graduate() {};

void set(string a, string b, string c, float g, int ag);

void print();

};

**Implementation:**

#include<iostream>

#include<string>

using namespace std;

#include"interface.h"

void faculty::print()

{

cout << first\_name << " " << last\_name << " with tel " << tel\_no << " - " << no\_of\_courses;

}

void faculty::set(string a, string b, int ag, int ex)

{

first\_name = a;

last\_name = b;

no\_of\_courses = ag;

tel\_no = ex;

}

void undergraduate::set(string a, string b, string c, float g, int ag)

{

age = ag;

cgpa = g;

fyp\_name = c;

first\_name = a;

last\_name = b;

}

void undergraduate::print()

{

cout << first\_name << " " << last\_name << " with gpa " << cgpa << " and age " << age << " has fyp name as " << fyp\_name;

}

void graduate::print()

{

cout << first\_name << " " << last\_name << " with gpa " << cgpa << " and age " << age << " has thesis name as " << thesis;

}

void graduate::set(string a, string b, string c, float g, int ag)

{

age = ag;

cgpa = g;

thesis = c;

first\_name = a;

last\_name = b;

}

**Driver:**

#include<iostream>

#include<string>

using namespace std;

#include "interface.h";

int main()

{

student objs;

graduate objg;

undergraduate obju;

faculty objf;

int num = 0,ag=0,extension=0,co=0;

float g=0;

string a, b,c;

cout << "who's data do you want to enter : n \n 1.grad student \n 2. undergrad student \n 3. faculty \n";

cin >> num;

switch (num)

{

case 1:

{

cout << "enter no. of students you want to enter : ";

cin >> co;

for (int i = 0; i < co; i++)

{

cout << "enter first name of student : ";

cin >> a;

cout << "enter last name of student : ";

cin >> b;

cout << "enter thesis name : ";

cin >> c;

cout << "enter gpa : ";

cin >> g;

cout << "enter age : ";

cin >> ag;

cout << endl;

objg.set(a, b, c, g, ag);

objg.print();

cout << endl;

}

break;

}

case 2:

{

cout << "enter no. of students you want to enter : ";

cin >> co;

for (int i = 0; i < co; i++)

{

cout << "enter first name of student : ";

cin >> a;

cout << "enter last name of student : ";

cin >> b;

cout << "enter fyp name : ";

cin >> c;

cout << "enter gpa : ";

cin >> g;

cout << "enter age : ";

cin >> ag;

cout << endl;

obju.set(a, b, c, g, ag);

obju.print();

cout << endl;

}

break;

}

case 3:

{

cout << "enter no. of faculty members you want to enter : ";

cin >> co;

for (int i = 0; i < co; i++)

{

cout << "enter first name of student : ";

cin >> a;

cout << "enter last name of student : ";

cin >> b;

cout << " enter no. of courses : ";

cin >> ag;

do

{

cout << " enter extension of 3 digits : ";

cin >> extension;

} while (extension >= 1000 && extension <= 99);

cout << endl;

objf.set(a, b, ag, extension);

objf.print();

cout << endl;

}

break;

}

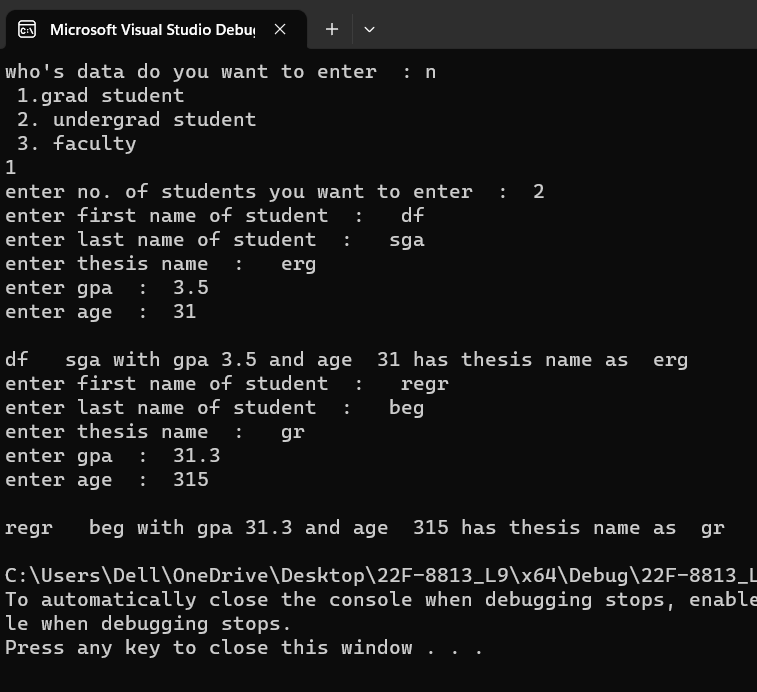
default:

cout << " wrong input ";

break;

}

}



Task 2:

///driver

#include<iostream>

#include<string>

#include"Class.h"

using namespace std;

int main()

{

string a, b, c;

static int num = 0;

persondata ob[20];

cout << "how many customers you want to enter data ? : ";

cin >> num;

customerdata o[20];

for (int i = 0; i < num; i++)

{

o[i].input();

cout << " firstname : ";/////////////////i haven't use getline in this because it give some blunders or like functioning in one and not in other...

cin >> a ;

cout << "\n lastname : ";/////////////////i haven't use getline in this because it give some blunders or like functioning in one and not in other...

cin >>b ;

cout << "\n address: ";

cin >> c; /////////////////i haven't use getline in this because it give some blunders or like functioning in one and not in other...

ob[i].setname(a, b, c);

}

for (int i = 0; i < num; i++)

{

ob[i].getname();

o[i].display();

}

}

//implementation

#include "Class.h"

void persondata::setname(string a, string b, string c)

{

firstname = a;

lastname = b;

address = c;

}

void persondata::getname()

{

cout << firstname << " " << lastname << " \n" << address << endl;

}

void customerdata::display()

{

cout << customernumber << " " << mailinglist<<endl;

}

void customerdata::input()

{

cout << " enter custom number : ";

cin >> customernumber;

cout << " \n enter 1. for true mailing list otherwise any num : ";

cin >> num;

if (num = 1)

{

mailinglist = true;

}

else

{

mailinglist = false;

}

}

//header

#pragma once

#ifndef CLASS\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS\_H

class persondata

{

private:

string firstname;

string lastname;

string address;

public:

persondata() {};

~persondata() {};

void setname(string a, string b, string c);

void getname();

};

class customerdata :public persondata

{

private:

int customernumber;

int num;

bool mailinglist = true;

public:

customerdata() {};

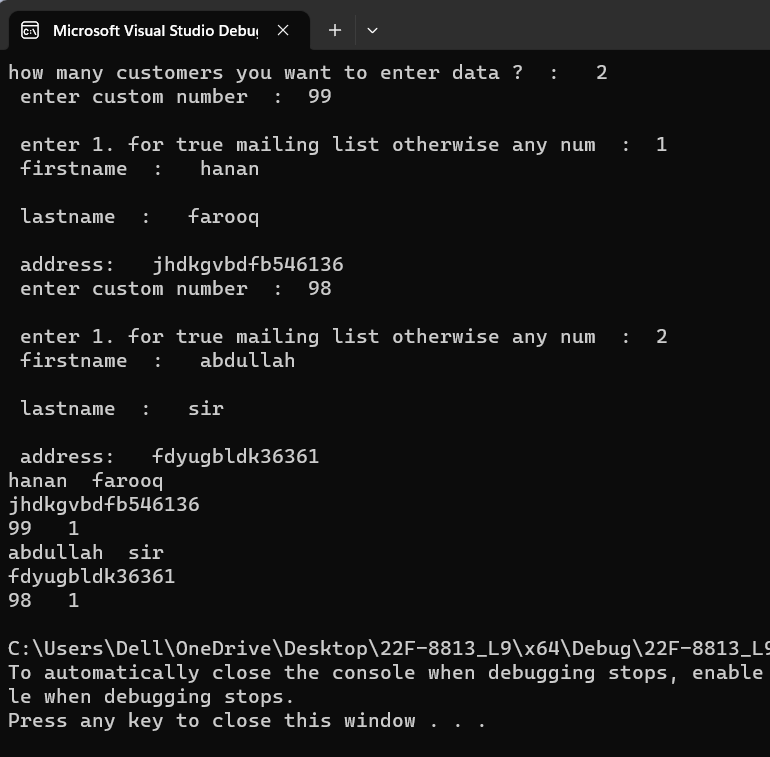
~customerdata() {};

void input();

void display();

};

#endif // !CLASS.H



Task 3:

///driver

#include<iostream>

#include<string>

#include"interface3.h"

using namespace std;

int main()

{

PreferredCustomer customer[100];

int num = 0;

cout << "enter number of customers you want to enter data : ";

cin >> num;

for (int i = 0; i < num; i++)

{

customer[i].getCustomerData();

double purchasesAmount;

cout << "Enter total purchases amount: ";

cin >> purchasesAmount;

customer[i].setPurchasesAmount(purchasesAmount);

customer[i].getDiscountLevel();

cout << "Customer Data:" << endl;

customer[i].OutputCustomerData();

customer[i].displaycustomerdiscount();

}

}

//implementation

#include "interface3.h"

// cpp file

#include<iostream>

#include<string>

using namespace std;

string PersonData::getFirstName() {

return FirstName;

}

void PersonData::setFirstName(string first) {

FirstName = first;

}

string PersonData::getLastName() {

return LastName;

}

void PersonData::setLastName(string last) {

LastName = last;

}

string PersonData::getAddress() {

return Address;

}

void PersonData::setAddress(string address) {

Address = address;

}

int CustomerData::getCustomerNumber() {

return CustomerNumber;

}

void CustomerData::setCustomerNumber(int customerNumber) {

CustomerNumber = customerNumber;

}

bool CustomerData::getMailingList() {

return MailingList;

}

void CustomerData::setMailingList(bool mailingList) {

MailingList = mailingList;

}

void CustomerData::getCustomerData() {

cout << "Enter customer number: ";

cin >> CustomerNumber;

cout << "Enter first name: ";

getline(cin.ignore(), FirstName);

cout << "Enter last name: ";

getline(cin, LastName);

cout << "Enter address: ";

getline(cin, Address);

cout << "Do you want to be on the mailing list? (1 for Yes / 0 for No): ";

cin >> MailingList;

}

void CustomerData::OutputCustomerData() {

cout << "Customer Number: " << CustomerNumber << endl;

cout << "First Name: " << FirstName << endl;

cout << "Last Name: " << LastName << endl;

cout << "Address: " << Address << endl;

if (MailingList) {

cout << "Mailing List: Yes" << endl;

}

else {

cout << "Mailing List: No" << endl;

}

}

PreferredCustomer::PreferredCustomer() : CustomerData() {

purchasesAmount = 0.0;

discountLevel = 0.0;

}

void PreferredCustomer::setPurchasesAmount(double purchases) {

purchasesAmount = purchases;

applyDiscount();

}

double PreferredCustomer::getPurchasesAmount() {

return purchasesAmount;

}

double PreferredCustomer::getDiscountLevel() {

return discountLevel;

}

void PreferredCustomer::applyDiscount() {

if (purchasesAmount >= 2000) {

discountLevel = 0.1;

}

else if (purchasesAmount >= 1500) {

discountLevel = 0.07;

}

else if (purchasesAmount >= 1000) {

discountLevel = 0.06;

}

else if (purchasesAmount >= 500) {

discountLevel = 0.05;

}

else {

discountLevel = 0.0;

}

}

void PreferredCustomer::displaycustomerdiscount()

{

cout << " the discount level is : " << discountLevel;

}

//header

#pragma once

#ifndef CLASS3\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS3\_H

#endif // !CLASS3.H

class PersonData {

protected:

string FirstName, LastName, Address;

public:

string getFirstName();

void setFirstName(string first);

string getLastName();

void setLastName(string last);

string getAddress();

void setAddress(string address);

};

class CustomerData : public PersonData {

private:

int CustomerNumber = 0;

bool MailingList = true;

public:

int getCustomerNumber();

void setCustomerNumber(int customerNumber);

bool getMailingList();

void setMailingList(bool mailingList);

void getCustomerData();

void OutputCustomerData();

};

class PreferredCustomer : public CustomerData {

private:

double purchasesAmount = 0.0, discountLevel = 0.0;

public:

PreferredCustomer();

PreferredCustomer(double purchases);

void setPurchasesAmount(double purchases);

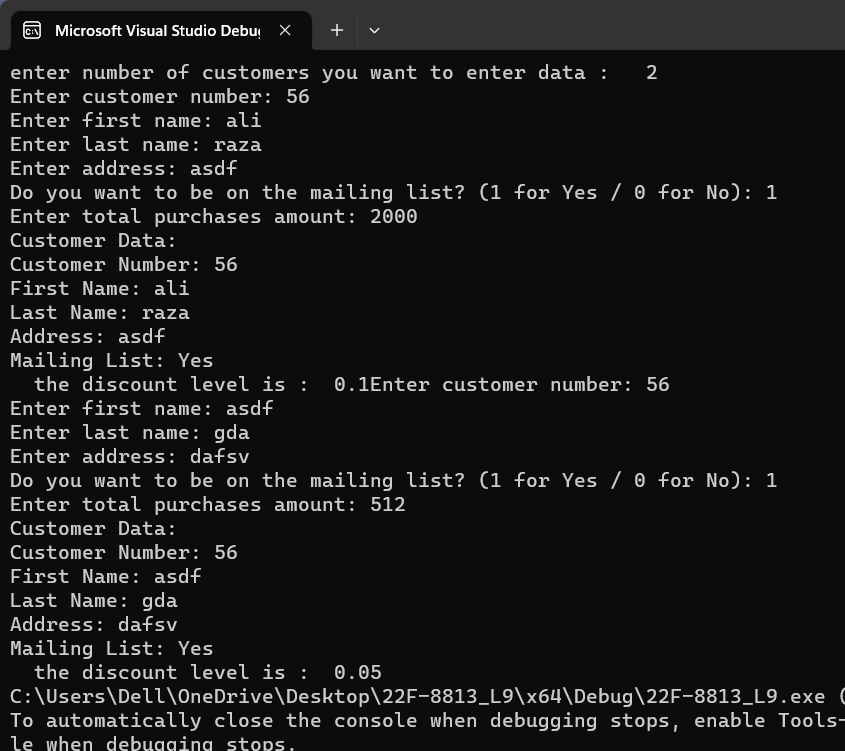
double getPurchasesAmount();

double getDiscountLevel();

void applyDiscount();

void displaycustomerdiscount();

};



Task 4:

//header

#pragma once

//#ifndef CLASS4\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS4\_H

//#endif // !CLASS3.H

class person

{

private:

int age;

string gender;

public:

string name;

person()

{

age = 19;

gender = "male";

}

~person() {};

};

class employed:public person

{

public:

employed() {};

string nic\_num;

int eage ;

string ename ;

void employ();

~employed() {};

};

class unemployed :public person

{

public:

int uage;

string uname ;

unemployed() {};

~unemployed() {};

void print();

};

class businessman:public employed

{

public:

string bname;

int bage;

businessman() {};

~businessman() {};

void display();

};

///driver

#include<iostream>

#include<string>

#include"interface4.h"

using namespace std;

int main()

{

businessman obj;

unemployed obju;

obj.name = "fawwaz";

obj.nic\_num = "32502-13513531-4";

obj.eage = 42;

obj.ename = "umar";

obju.uage = 17;

obju.uname = "haider";

obj.bname = "ali";

obj.bage = 37;

obj.employ();

obj.display();

obj.name = "asc";

obju.print();

}

//implementation

#include "interface4.h"

void employed::employ()

{

cout << "“Hi, I am Employ from Employed Class";

}

void businessman::display()

{

cout << "person name is : " << name;

cout << "\nbusiness man name is : " << bname << " and age is : " << bage;

cout << "\nemployed man name is : " << ename << " and age is : " << eage << " with NIC no. is : " << nic\_num;

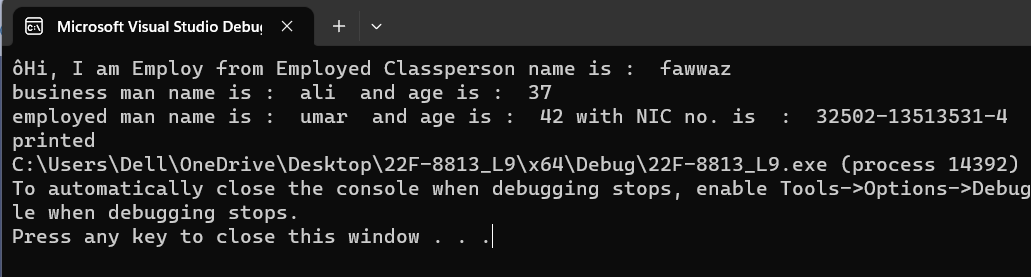
}

void unemployed::print()

{

cout << "\nprinted";

}



Task 5:

//header

#pragma once

//#ifndef CLASS5\_H

#include<string>

#include<iostream>

using namespace std;

#define CLASS5\_H

//#endif // !CLASS5.H

class vehicle

{

private:

int speed;

int distance;

float duration;

public:

vehicle(int y = 0) :speed(y), distance(y), duration(y) {};

void setspeed(int sped) { speed = sped; };

int getspeed() { return speed; };

void setduration(float durat) { duration= durat; };

float getduration() { return duration; };

void setdistance(int dist) { distance = dist; };

int getdistance() { return distance; };

~vehicle() {};

};

class wheelvehicle :public vehicle

{

private:

int wheels;

public:

wheelvehicle(int x = 0) :wheels(x) {};

void setwheels(int whel) { wheels = whel; };

int getwheels() { return wheels; };

~wheelvehicle() {};

};

class wingvehicle :public vehicle

{

private:

int wings;

public:

void winput();

void wprint();

wingvehicle(int z = 0) :wings(z) {};

void setwings(int wing) { wings = wing; };

int getwings() { return wings; };

~wingvehicle() {};

};

class truck :public wheelvehicle

{

private:

int carryingload;

public:

void input();

truck(int a = 0) :carryingload(a) {};

void setload(int load) { carryingload = load; };

int getload() { return carryingload; };

~truck() {};

void print();

};

//implementation

#include "interface5.h"

void truck::input()

{

int sped=0, dist=0, whel=0, carryload = 0;

float durat = 0;

cout << "enter speed : "; cin >> sped;

cout << "enter distance : "; cin >> dist;

cout << "enter wheel : "; cin >> whel;

cout << "enter carryingload: "; cin >> carryload;

setdistance(dist);

setload(carryload);

setspeed(sped);

setwheels(whel);

durat = dist\*1.0 / sped\*1.0;

setduration(durat);

}

void truck::print()

{

cout << " \nspeed is : " << getspeed();

cout << " \nload is : " << getload();

cout << " \nduration is : " << getduration();

cout << " \ndistance is : " << getdistance();

cout << " \nwheels is : " << getwheels();

}

void wingvehicle::winput()

{

int wing = 0;

cout << "enter wing : "; cin >> wing;

setwings(wing);

}

void wingvehicle::wprint()

{

cout << "\n wings are : " << getwings();

}

///driver

#include<iostream>

#include<string>

#include"interface5.h"

using namespace std;

int main()

{

int num = 0;

truck obj[100];

wingvehicle wobj;

cout << "enter no. of vehicles you want to enter data : "; cin >> num;

for (int i = 0; i < num; i++)

{

obj[i].input();

wobj.winput();

}

for (int i = 0; i < num; i++)

{

obj[i].print();

wobj.wprint();

}

}

