QUESTION NO 1:

#include<iostream>

#include<string>

#include<cstring>

using namespace std;

class student

{

protected:

string name;

string lastname;

int year;

int id;

string pos;

public:

student()

{

this->name = "Saad";

this->lastname = "Ashraf";

this->year = 21;

this->id = 9167;

pos = "2";

}

void initalize()

{

cout << "Input the Name : ";

cin >> name;

cout << "Input the Father Name : ";

cin >> lastname;

cout << "Input the Batch : ";

cin >> year;

cout << "Input the Roll Number : ";

cin >> id;

cout << "Input the Society : ";

cin >> pos;

}

void print()

{

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

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

cout << "Batch Number : " << year <<endl;

cout << "Roll Number : " << id << endl;

cout << "Society Position : " << pos <<endl;

}

};

class Participant :public student

{

private:

int number;

int wins;

string type;

string\* X;

string\* Y;

int\* batchnum;

int\* rol;

string\* A;

public:

Participant()

{

this->number = 012;

cout << "Input the number of Particpants : ";

cin >> number;

this->wins = 212;

this->type = "Position";

X = new string[number];

Y = new string[number];

batchnum = new int[number];

rol = new int[number];

A = new string[number];

}

void input()

{

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

{

cout << "Enter the data of participant " << i + 1 << " : " << endl;

student::initalize();

cout << "Enter the number of privious wins : "; cin >> wins;

cout << "Enter the category the participant wants to choose : ";

cin >> type;

X[i] = name;

Y[i] = lastname;

batchnum[i] = year;

rol[i] = id;

A[i] = pos;

cout << endl;

}

}

void display()

{

cout << "Number of Participants are/is " << number << endl;

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

{

cout << endl;

cout << "INFO of Participant " << i + 1 << " is as follows :" << endl;

cout << "Name : " << X[i] << endl;

cout << "Last Name : " << Y[i] << endl;

cout << "Batch : " << batchnum[i] << endl;

cout << "Roll Number : " << rol[i] << endl;

cout << "Society Position : " << A[i] << endl;

cout << "--------------------------";

}

}

};

class TeamHead :public student

{

private:

string teams;

public:

TeamHead()

{

teams = "xyz";

}

void Tinput()

{

student::initalize();

cout << "Input the name of the team : ";

cin >> teams;

}

void Display2()

{

student::print();

cout << "Team Name : " << teams << endl;

}

};

int main()

{

student s;

Participant var;

TeamHead var1;

cout << endl << "Enter the Information of People" << endl;

var.input();

cout << endl << "Enter the Information of Team leader" << endl;

var1.Tinput();

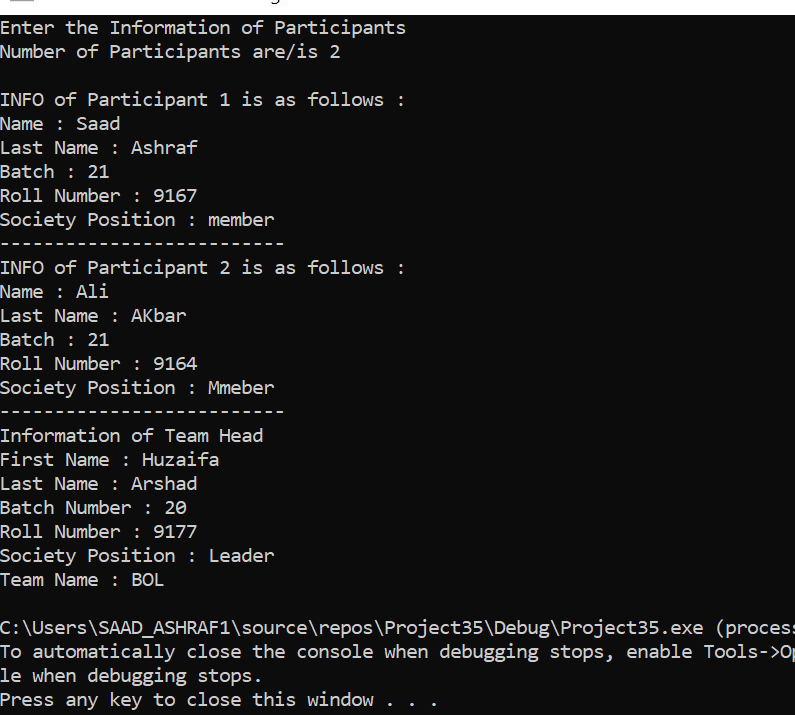
cout << endl << "Enter the Information of Participants" << endl;

var.display();

cout << endl << "Information of Team Head" << endl;

var1.Display2();

}



QUESTION NO 2:

#include <iostream>

#include <string>

using namespace std;

class PersonData

{

private:

string city;

string address;

string lastName;

string firstName;

string zip;

string state;

string phone;

public:

void initalize()

{

cin >> firstName;

cin >> lastName;

cin >> address;

cin >> city;

cin >> state;

cin >> zip;

cin >> phone;

}

string getLastName()

{

return lastName;

}

void setLastName(string lastName)

{

this->lastName = lastName;

}

string getFirstName()

{

return firstName;

}

void setFirstName(string firstName)

{

this->firstName = firstName;

}

string getAddress()

{

return address;

}

void setAddress(string address)

{

this->address = address;

}

string getCity()

{

return city;

}

void setCity(string city)

{

this->city = city;

}

string getState()

{

return state;

}

void setState(string state)

{

this->state = state;

}

string getZip()

{

return zip;

}

void setZip(string zip)

{

this->zip = zip;

}

string getPhone()

{

return phone;

}

void setPhone(string phone)

{

this->phone = phone;

}

};

class CustomerData : public PersonData

{

private:

int customerNumber;

bool mailingList;

public:

void initalize2() {

cin >> customerNumber;

cin >> mailingList;

}

int getCustomerNumber()

{

return customerNumber;

}

void setCustomerNumber(int customerNumber)

{

this->customerNumber = customerNumber;

}

bool getMailingList()

{

return mailingList;

}

void setMailingList(int mailingList)

{

this->mailingList = mailingList;

}

void DisplayCustomerData() {

cout <<"Name : " << getFirstName() <<endl;

cout << "Last Name :" <<getLastName() <<endl;

cout <<"Address : " << getAddress() <<endl;

cout <<"City : " << getCity() <<endl;

cout <<"State : "<< getState() <<endl;

cout <<"Zip : "<< getZip() <<endl;

cout <<"Phone Number : "<< getPhone() <<endl;

cout <<"Customer NO: " << getCustomerNumber() <<endl;

cout << "Mailing List : " << getMailingList();

}

};

int main()

{

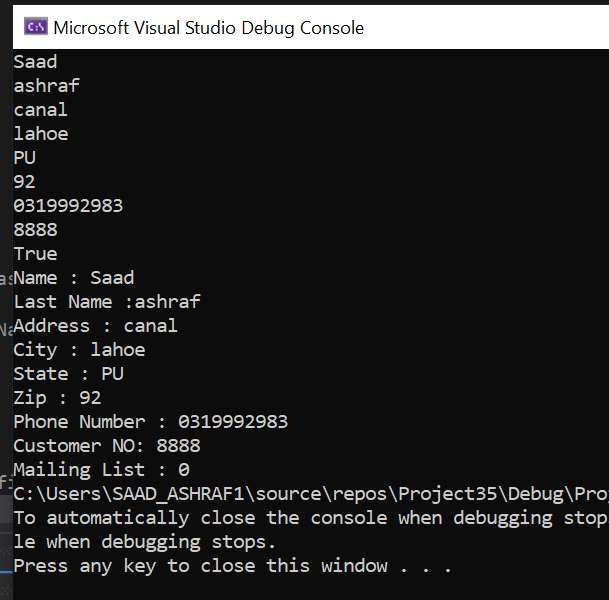
CustomerData customer;

customer.initalize();

customer.initalize2();

customer.DisplayCustomerData();

}



QUESTION NO 5:

Constant Functions means that they will remain same throughout the execution of the program. Non-constant variable are those whose value can be changed at any point during the execution of the program. Therefore, Non constant variables can't be used because they change the behaviour of the function

#include <iostream>

using namespace std;

class Test {

    int value;

public:

    Test(int v = 0) { value = v; }

    int getValue() const { return value; }

};

int main()

{

    Test t(20);

    cout << t.getValue();

    return 0;

}

QUESTION NO 6:

#include<iostream>

using namespace std;

class Time {

private:

int hours;

int minutes;

int seconds;

public:

Time() { //constructor to initalize with 0

hours = 0;

minutes = 0;

seconds = 0;

};

Time(int hours, int minutes, int seconds) {

this->hours = hours; //using this pointer

this->minutes = minutes;

this->seconds = seconds;

};

int hoursfun() {

return hours;

};

int minutesfun() {

return minutes;

};

int secondsfun() {

return seconds;

};

void display() { //function to display output

cout << hours << ":" << minutes << ":" << seconds << endl;

};

Time add(Time x, Time y) {

int add = x.hoursfun() + y.hoursfun(); //adding the hours

if (23 < add) {

add -= 24;

}

int minadd = x.minutesfun() + y.minutesfun(); //adding the minutes

if (59 < minadd) { //logic to increase hours if minutes is greater than or equal to 60

minadd -= 60;

add += 1;

}

int secadd = x.secondsfun() + y.secondsfun(); //adding the seconds

if (secadd > 59) { //logic to increase minute if seconds is greater than or equal to 60

secadd -= 60;

minadd += 1;

}

Time time3(add, minadd, secadd);

return time3;

};

};

int main() {

Time t1(13, 44, 39); //initalizing with fixed values

Time t2(19, 17, 55); //initalizing with fixed values

Time t3;

t3 = t3.add(t1, t2);

t1.display();

t2.display();

t3.display();

return 0;

}

