**Question#1**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class Student {

private:

string name;

const string cnic;

const char gender;

float cgpa;

public:

Student(string Name, string CNIC, char Gender, float CGPA) : name(Name), cnic(CNIC), gender(Gender), cgpa(CGPA) {}

void printData() {

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

cout << "Student CNIC :" << cnic << endl;

cout << "Gender : " << gender << endl;

cout << "CGPA : " << cgpa << endl;

}

void setCgpa(float CGPA) {

cgpa = CGPA;

}

};

class Section {

private:

Student \*students[40];

string sectionName;

string teacher;

int n;

public:

Section() {

sectionName = " ";

teacher = " ";

n = 0;

}

Section(string name,string Teacher) {

sectionName = name;

teacher = Teacher;

n = 0;

}

void editSection(string name, string Teacher) {

sectionName = name;

teacher = Teacher;

}

void addStudent(Student \*obj) {

if (n < 40) {

students[n] = obj;

n = n + 1;

}

}

void updateStudent(int index,float cgpa) {

if (index >= 0 && index < n) {

students[index]->setCgpa(cgpa);

}

}

void printList() {

cout << "Section: " << sectionName << endl;

cout << "Teacher: " << teacher << endl;

cout << "Students:" << endl;

for (int i = 0; i < n; ++i) {

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

students[i]->printData();

cout << endl;

}

}

void printSectionList() {

for (int i = 0; i < n; i++) {

cout << "Section Name : " << sectionName << endl;

cout << "Teacher Name : " << teacher << endl;

}

}

};

int main() {

int n = 0, choice = 0;

float cgpa = 0;

cout << "Enter Total Number of Sections : "; cin >> n;

Section\* section = new Section[n];

string sectionName;

string teacher;

for (int i = 0; i < n; i++) {

cout << "Enter The Name of Section : "; cin >> sectionName;

cout << "Enter The Name of Teacher of The Section : "; cin >> teacher;

section[i] = Section(sectionName, teacher);

}

cout << "Enter 1 to Edit Section Attributes" << endl;

cout << "Enter 2 to Add Student in The Section" << endl;

cout << "Enter 3 to Update Student of a Section" << endl;

cout << "Enter 4 to Print List of Students in a Section" << endl;

cout << "Enter 5 to Print List of Sections" << endl;

cout << "Enter 6 to Exit" << endl;

cout << "Enter Your Choice : "; cin >> choice;

do {

switch (choice) {

case 1:

{

cout << "Enter Section Number : "; cin >> n;

cout << "Enter New Section Name : "; cin >> sectionName;

cout << "Enter New Teacher Name : "; cin >> teacher;

section[n - 1].editSection(sectionName, teacher);

break;

}

case 2:

{

char gender;

string cnic, name;

cout << "Enter Section Number : "; cin >> n;

cout << "Enter Student Name : "; cin >> name;

cout << "Enter Student CNIC : "; cin >> cnic;

cout << "Enter Gender : "; cin >> gender;

cout << "Enter CGPA : "; cin >> cgpa;

section[n - 1].addStudent(new Student(name, cnic, gender, cgpa));

break;

}

case 3:

{

int temp = 0;

cout << "Enter Section Number : "; cin >> n;

cout << "Enter Student Number : "; cin >> temp;

cout << "Enter CGPA : "; cin >> cgpa;

section[n - 1].updateStudent(temp, cgpa);

break;

}

case 4:

{

cout << "Enter Section Number : "; cin >> n;

section[n - 1].printList();

break;

}

case 5:

{

cout << "List of Sections : " << endl;

section->printSectionList();

break;

}

case 6:

{

cout << "Exiting........" << endl;

break;

}

default:

cout << "Invalid Input" << endl;

}

} while (choice != 6);

delete[] section;

return 0;

}

**Output:**

****

**Question#2**

**Code:**

#include<iostream>

using namespace std;

class Jagged {

private:

int row, col;

int\*\* arr;

public:

Jagged() {

cout << "Constructor Called" << endl;

row = 0;

col = 0;

arr = nullptr;

}

Jagged input() {

cout << "Enter Number of Rows : "; cin >> row;

arr = new int\*[row];

for (int i = 0; i < row; i++) {

cout << "Enter Number of Columns For Row#" << i + 1 << " : "; cin >> col;

arr[i] = new int[col + 1];

arr[i][0] = col;

}

for (int i = 0; i < row; i++) {

int temp = arr[i][0];

for (int j = 1; j <= temp; j++) {

cout << "Enter Value For Row# " << i + 1 << " Column#" << j << " : "; cin >> arr[i][j];

}

}

return \*this;

}

void display(Jagged &obj) {

for (int i = 0; i < row; i++) {

int temp = arr[i][0];

for (int k = 0; k < 40; k++) {

cout << "-";

}

cout << endl;

for (int j = 1; j <= temp; j++) {

cout << "| " << arr[i][j] << " | ";

}

cout << endl;

for (int k = 0; k < 40; k++) {

cout << "-";

}

cout << endl;

}

}

};

int main() {

Jagged obj;

obj.input();

obj.display(obj);

return 0;

}

**Output:**

****

**Question#3**

**Code:**

**Part(a):**

#include<iostream>

using namespace std;

class CountClass {

private:

static int count;

public:

static void counter() {

count = count + 1;

}

CountClass() {

counter();

}

static void display() {

cout << "No of Instances : " << count << endl;

}

};

int CountClass::count = 0;

int main() {

CountClass obj[10];

obj[0].display();

return 0;

}

**Part(b):**

#include<iostream>

using namespace std;

class Example

{

public:

Example(int y = 10) : data(y) {}

//end Example constructor int getIncrementedData() const

int getIncrementedData()

{

return ++data;

}

//end function getIncrementedData static int getCount()

void display\_data\_before\_increment()

{

cout << "Data is " << data << endl;

}

static int get\_count()

{

return count;

} //end function getCount

private:

int data;

static int count;

}; // end class Example

int Example::count = 0;

int main()

{

Example obj;

obj.display\_data\_before\_increment();

cout << "incremented data:" << obj.getIncrementedData() << endl;

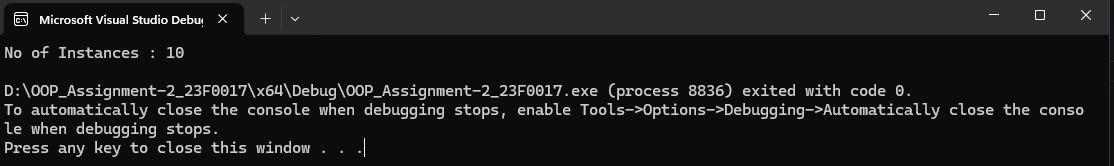
cout << "count:" << Example::get\_count();

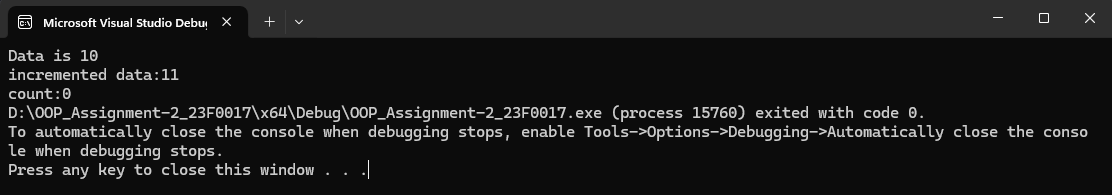
return 0;

}

**Output:**

**Part(a):**

****

**Part(b):** ****

**Question#4**

**Code:**

#include<iostream>

#include<conio.h>

using namespace std;

class tollBooth {

private:

unsigned int noOfCars;

double totalAmount;

public:

tollBooth() {

cout << "Constructor Called" << endl;

noOfCars = 0;

totalAmount = 0;

}

void payingCar() {

noOfCars = noOfCars + 1;

totalAmount = totalAmount + 0.50;

}

void nopayCar() {

noOfCars = noOfCars + 1;

}

void display() const {

cout << "Number of Car Passed : " << noOfCars << endl;

cout << "Total Amount Collected : $" << totalAmount << endl;

}

};

int main() {

tollBooth obj;

char input;

cout << "Press 1 For a Paying Car" << endl;

cout << "Press 2 For a Non-Paying Car" << endl;

do {

input = \_getch();

switch (input) {

case '1':

cout << "Paying Car Counted" << endl;

obj.payingCar();

break;

case '2':

cout << "Non Paying Counted" << endl;

obj.nopayCar();

break;

case 27:

break;

default:

cout << "Invalid Input" << endl;

cout << "Press 1 For a Paying Car" << endl;

cout << "Press 2 For a Non-Paying Car" << endl;

}

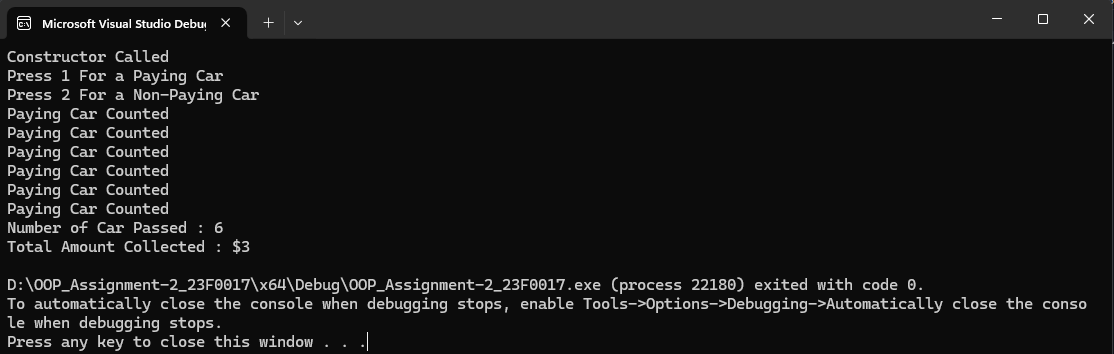
} while (input != 27);

obj.display();

return 0;

}

**Output:**

****

**Question#5**

**Code:**

#include<iostream>

using namespace std;

class Time {

private:

int hours, minutes, seconds;

public:

Time() {

cout << "Default Constructor Called" << endl;

hours = 0;

minutes = 0;

seconds = 0;

}

Time(int h, int m, int s) {

hours = h;

minutes = m;

seconds = s;

}

~Time() {

cout << "Default Destructor Called" << endl;

}

void displayTime12() {

cout << "Time in 12-Hour Format : " << hours % 12 << " : " << minutes << " : " << seconds << endl;

}

Time addTime(const Time& t,const Time& t1) {

Time sum(0, 0, 0);

sum.hours = t.hours + t1.hours;

sum.minutes = t.minutes + t1.minutes;

sum.seconds = t.seconds + t1.seconds;

if (sum.seconds >= 60) {

seconds = seconds % 60;

sum.minutes = sum.minutes + 1;

}

if (sum.minutes >= 60) {

sum.minutes = sum.minutes % 60;

sum.hours = sum.hours + 1;

}

if (sum.hours >= 24) {

sum.hours = sum.hours % 24;

}

return sum;

}

};

int main() {

cout << "Time Object#1 : " << endl;

Time t(1, 2, 3);

t.displayTime12();

cout << "Time Object#2 : " << endl;

Time t1(4, 5, 6);

t1.displayTime12();

cout << "Time Object#3 : " << endl;

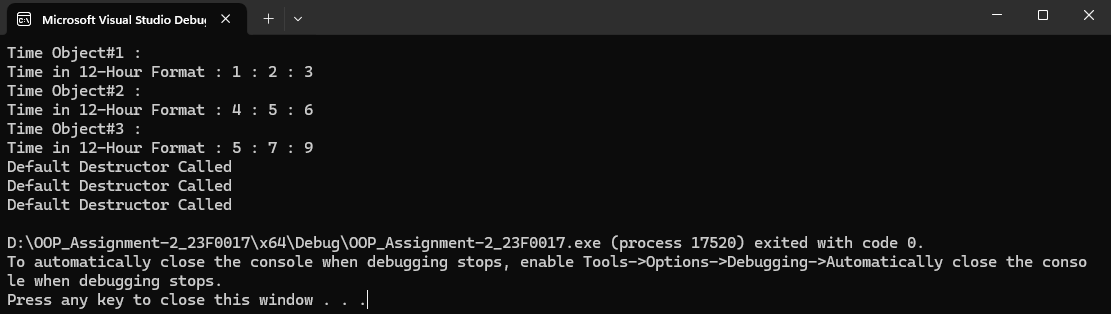
Time t2 = t2.addTime(t, t1);

t2.displayTime12();

return 0;

}

**Output:**

****

**Question#6**

**Code:**

**.h file:**

class TestScores {

private:

int\* score;

int size;

double avg;

public:

TestScores(int n);

TestScores(TestScores& obj);

~TestScores();

TestScores& operator=(const TestScores& obj);

void setScore();

void getScore();

double Average();

};

**Class.cpp:**

#include "Q6.h"

#include <iostream>

using namespace std;

TestScores :: TestScores(int n) {

avg = 0;

size = n;

score = new int[size];

}

TestScores::TestScores(TestScores& obj) {

size = obj.size;

score = new int[size];

for (int i = 0; i < size; ++i) {

score[i] = obj.score[i];

}

avg = obj.avg;

}

TestScores :: ~TestScores() {

delete[] score;

}

void TestScores::setScore() {

for (int i = 0; i < size; i++) {

cout << "Enter Score #" << i + 1 << " : "; cin >> score[i];

}

}

void TestScores:: getScore() {

for (int i = 0; i < size; i++) {

cout << "Score #" << i + 1 << " : " << score[i] << endl;

}

}

double TestScores :: Average() {

double sum = 0;

for (int i = 0; i < size; i++) {

sum = sum + score[i];

}

avg = sum / size;

return avg;

}

TestScores& TestScores::operator=(const TestScores& obj) {

if (this == &obj) {

return \*this;

}

size = obj.size;

delete[] score;

score = new int[size];

for (int i = 0; i < size; i++) {

score[i] = obj.score[i];

}

avg = obj.avg;

return \*this;

}

**Main.cpp:**

#include "Q6.h"

#include <iostream>

using namespace std;

int main() {

int n = 0;

cout << "Enter Number of Test Scores You Want to Enter : "; cin >> n;

TestScores obj(n);

obj.setScore();

obj.getScore();

cout << "Average Score For Object#1 : " << obj.Average() << endl;

TestScores obj1(obj);

cout << "Object#2" << endl;

obj1.getScore();

cout << "Average Score For Object#2 : " << obj1.Average() << endl;

TestScores obj2 = obj1;

cout << "Object#3" << endl;

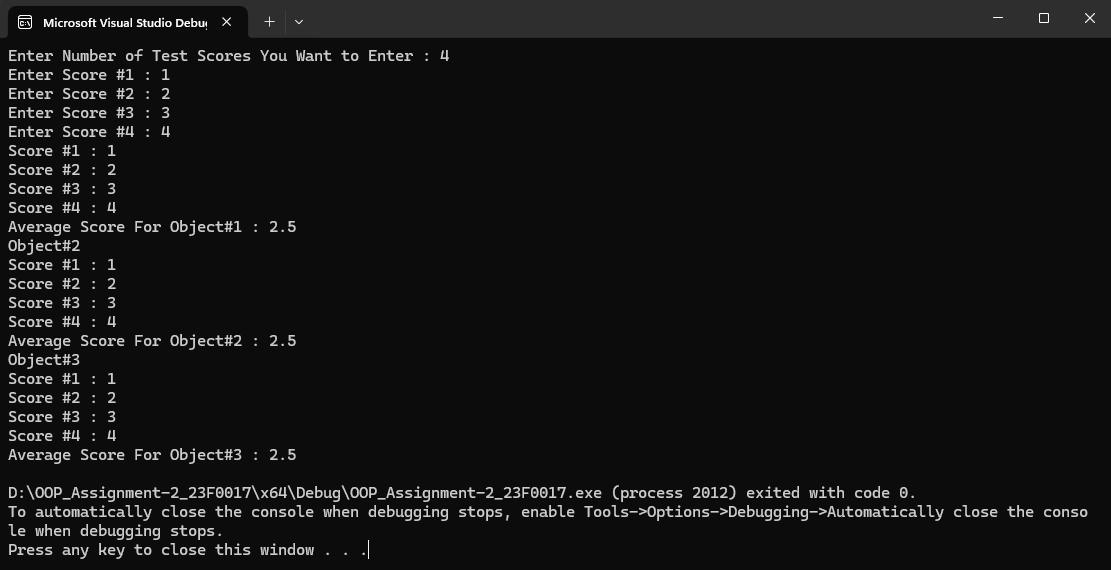
obj2.getScore();

cout << "Average Score For Object#3 : " << obj2.Average() << endl;

return 0;

}

**Output:**

****

**Question#7**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class Car {

private:

string carName;

int model;

string color;

string plateNumber;

public:

Car(string arr,int var,string arr1,string arr2) {

carName = arr;

model = var;

color = arr1;

plateNumber = arr2;

}

Car(Car& obj) {

carName = obj.carName;

model = obj.model;

color = obj.color;

plateNumber = obj.plateNumber;

}

void display() {

cout << "Name : " << carName << endl;

cout << "Plate Number : " << plateNumber << endl;

cout << "Model : " << model << endl;

cout << "Color : " << color << endl;

}

};

int main() {

string carName;

int model;

string color;

string plateNumber;

cout << "Enter Car Name : "; cin >> carName;

cout << "Enter Model : "; cin >> model;

cout << "Enter Color : "; cin >> color;

cout << "Enter Plate Number : "; cin >> plateNumber;

Car obj1(carName, model, color, plateNumber);

Car obj(obj1);

cout << "CAR #1 : " << endl;

obj.display();

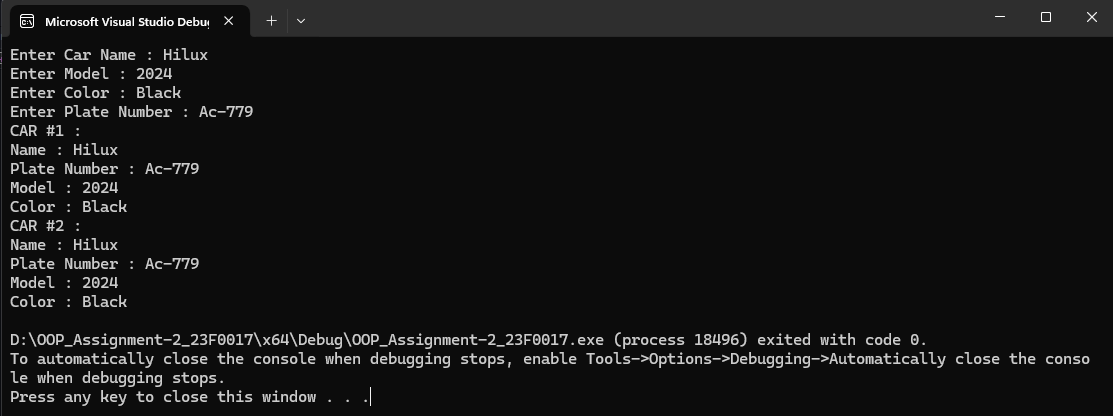
cout << "CAR #2 : " << endl;

obj1.display();

return 0;

}

**Output:**

****

**Question#8**

**Code:**

#include<iostream>

#include<string>

using namespace std;

class Scientist {

private:

int ID;

string name;

string DOB;

string graduatedFrom;

string pickedDomain;

public:

Scientist() {

ID = 0;

name = " ";

DOB = " ";

graduatedFrom = " ";

pickedDomain = " ";

}

void InputData(int id, string Name, string dob, string GraduatedFrom, string PickedDomain) {

ID = id;

name = Name;

DOB = dob;

graduatedFrom = GraduatedFrom;

pickedDomain = PickedDomain;

}

void DisplayData() {

cout << "ID : " << ID << endl;

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

cout << "Date of Birth : " << DOB << endl;

cout << "Graduated From : " << graduatedFrom << endl;

cout << "Picked Domain : " << pickedDomain << endl;

}

};

class Chemist : public Scientist {

private:

string favouriteChemical;

string chemRank;

public:

Chemist() :Scientist() {

favouriteChemical = " ";

chemRank = " ";

}

void inputData(int id, string Name, string dob, string GraduatedFrom, string PickedDomain, string chemical, string rank){

InputData(id, Name, dob, GraduatedFrom,PickedDomain);

favouriteChemical = chemical;

chemRank = rank;

}

void displayData(){

DisplayData();

cout << "Favourite Chemical : " << favouriteChemical << endl;

cout << "Chemical Rank : " << chemRank << endl;

}

};

int main() {

Chemist obj;

obj.inputData(001, "XYZ" , "21/04/1889", "University", "Chemistry", "ABC", "IV");

obj.displayData();

return 0;

}**Output:**

