National University of Computer and Emerging Sciences



Lab Exercise 08

For

Object Oriented Programming Lab

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No | 19F0228 |
| Course Instructor(s) | Dr. Danish |
| Lab Instructor(s) | Mr. Mughees Ismail |
| Semester | Spring 2020 |

|  |
| --- |
| **Question#1** |

# Source code:

#include<iostream>

#include<string>

using namespace std;

class student {

string Name;

string CNIC;

char gender;

float CGPA;

static int count;// (that counts the numbers of students of a section)

public:

student()

{

Name = "--";

CNIC = "--";

gender = '-';

CGPA = 0.0;

count++;

}

void set\_name(string Name)

{

this->Name = Name;

}

void set\_Cnic(string CNIC)

{

this->CNIC = CNIC;

}

void set\_gender(char gender)

{

this->gender = gender;

}

float set\_Gpa(float CGPA)

{

this->CGPA = CGPA;

}

void set\_Student\_data(string CNIC)

{

cout << "Please enter the name of the student " << endl;

cin >> Name;

cout << "Please enter the gender" << endl;

cout << endl;

cout << "For male press M " << endl;

cout << "For female press F " << endl;

cout << "if you dont want to enter press S " << endl;

{

while (1)

{

cin >> gender;

if (gender == 'm' || gender == 'm' || gender == 'f' || gender == 'F' || gender == 's' || gender == 'S')

{

break;

}

else

{

cout << "Invalid input \n Please input again" << endl;

}

}

}

cout << "Enter student's CGPA:";

cin >> CGPA;

cout << "Cnic of the student is " << CNIC << endl;

}

void get\_Student\_data()

{

cout << Name << "\t\t" << CNIC << "\t\t" << gender << "\t\t" << CGPA << endl;

}

int counter()

{

return count;

}

};

int student::count = 0;

class section

{

student std[5];

char Section\_name;

string Class\_teacher;

static int Count; // (that count the number of sections created)

public:

section()//constructor

{

Section\_name = '-';

Class\_teacher = '-';

Count++;

}

void set\_section\_data()

{

cout << "Please input the name of the section" << endl;

cin >> Section\_name;

cout << "Please input the Name of the teacher" << endl;

cin >> Class\_teacher;

}

void editSection()

{

string cnic;

cout << "Input the section name to edit" << endl;

cin >> Section\_name;

cout << "Input the Cnic of the student to move " << endl;

cin >> cnic;

student std;//student calss ka local object

std.set\_Student\_data(cnic);

cout << "Input the next section name in which student will move " << endl;

cin >> Section\_name;

}

void addStudent()

{

string cnic;

int number\_of\_students = 0;

cout << "Input the Cnic of the student to move " << endl;

cin >> cnic;

cout << "Enter the section in which you want to add the student" << endl;

cin >> Section\_name;

cout << "How many students you want to enter" << endl;

cin >> number\_of\_students;

student \*add\_STD\_Pointer = new student[number\_of\_students];

add\_STD\_Pointer[Section\_name].set\_Student\_data(cnic);

}

void updateStudent()

{

string cnic;

cout << "Input the Cnic of the student you want to update" << endl;

cin >> cnic;

cout << "Enter the section in which you want to add the student" << endl;

cin >> Section\_name;

student \*add\_STD\_Pointer = new student[1];

add\_STD\_Pointer[Section\_name].set\_Student\_data(cnic);

}

char printList()//section +class teacher

{

cout << Section\_name << "\t\t" << Class\_teacher << endl;

}

static int printCount()

{

return Count;

//kitnay section hain

}

int Menu\_functionalities()

{

int number;

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

cout << "Press 2 to Add Student in a section" << endl;

cout << "Press 3 to Update Student of a section" << endl;

cout << "Press 4 to Print List of Students of a section" << endl;

cout << "Press 5 to Print List of Sections (can be done in main())" << endl;

cout << "Press 6 to Print Count of Sections and Students in each section" << endl;

cout << "Press 7 to quit" << endl;

cin >> number;

return number;

}

};

int section::Count = 0;

int main()

{

string c;//cnic

int no\_of\_students = 0;

int no\_of\_sections = 0;

int no\_of\_ibjects = 0;

cout << "How may sections you want to create ?" << endl;

cin >> no\_of\_sections;

cout << "How may students you want to put in 1 sections ?" << endl;

cin >> no\_of\_students;

section \*section\_ptr = new section[no\_of\_sections];

student \*student\_ptr = new student[no\_of\_sections];

cout << "You have to put data in sections." << endl;

for (int i = 0; i <no\_of\_sections; i++)

{

cout << "Section number:" << i + 1 << endl;

section\_ptr[i].set\_section\_data();

for (int j = 0; j < no\_of\_students; j++)

{

cout << "Student no." << j + 1 << endl;

cout << "Enter cnic" << endl;

cin >> c;

student\_ptr[j].set\_Student\_data(c);

system("cls");

}

}int number = 1;

section option;

while (number != 0)

{

int choice = option.Menu\_functionalities();

system("cls");

if (choice == 1)

{

section one;

one.editSection();

}

else if (choice == 2)

{

section two;

two.addStudent();

}

else if (choice == 3)

{

section three;

three.updateStudent();

}

else if (choice == 4)

{

cout << " Name\t\t CNIC\t\t gender<<\t\t CGPA " << endl;

student\_ptr[0].get\_Student\_data();

for (int i = 0; i < no\_of\_sections; i++)

{

student\_ptr[i].get\_Student\_data();

cout << endl;

}

}

else if (choice == 5)

{

//yahan sections print krnay

cout << "Section\t\tClass Teacher" << endl;

for (int i = 0; i < no\_of\_sections; i++)

{

section\_ptr[i].printList();

}

}

else if (choice == 6)

{

cout << "Total sections are:" << endl;

cout << section\_ptr[0].printCount() << endl;//section

cout << "Total students in each sections are:" << endl;

cout << (student\_ptr[0].counter()) / 2 << endl;

}

else if (choice == 7)

{

cout << "Thank you :-)" << endl;

break;

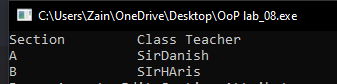
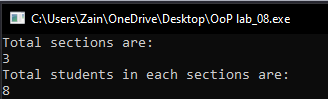
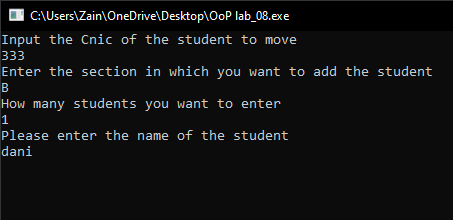
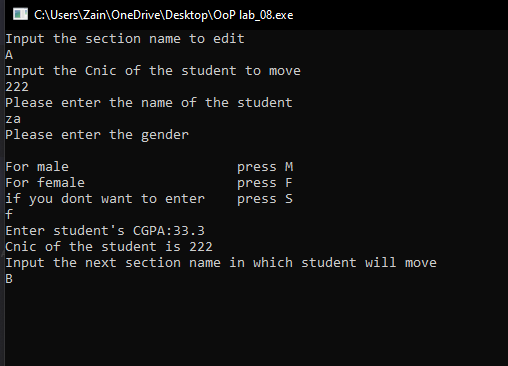
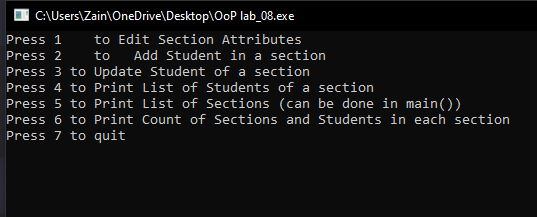
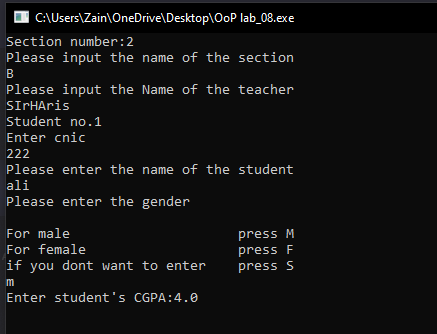
}

}

system("pause>0");

}

# Snip:



|  |
| --- |
| **Question#2** |

# Source Code:

#include<iostream>

#include<string>

using namespace std;

struct timer

{

int hours;

int minute;

int seconds;

};

class Testing

{

int test\_id;

string question;

timer Timer;

int static count;

Testing()

{

count++;

test\_id = 0;

question = "default";

Timer.hours = 00;

Timer.minute = 00;

Timer.seconds = 00;

}

public:

static Testing\* indirect\_approach(int size)

{

Testing \*pointer = new Testing[size];

return (pointer);

}

void setTestId(int test\_id)

{

this->test\_id = test\_id;

}

void setQuestion(string question)

{

this->question = question;

}

void setTimer()

{

cout << "Please input the time" << endl;

cin >> Timer.hours;

cin >> Timer.minute;

cin >> Timer.seconds;

}

int getCounter()

{

return count;

}

void display() {

cout << test\_id << " " << question << " " << Timer.hours << " : " << Timer.minute << " : " << Timer.seconds << endl;

;

}

};

int Testing::count = 0;

int main()

{

int uniq;

string q;

int number;

Testing \*ptr;

cout << "Please input the number of tests" << endl;

cin >> number;

ptr = Testing::indirect\_approach(number);

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

{

cout << "Please input the test id" << endl;

cin >> uniq;

ptr[i].setTestId(uniq);

ptr[i].setQuestion("This is the Question Number ");

ptr[i].setTimer();

system("cls");

}

cout << "tid\t\tquestion\t\t\t time" << endl;

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

{

ptr[i].display();

}

cout << endl << endl;

cout << "Total number of questions is:" << endl;

cout << ptr[number].getCounter();

system("pause>0");

}

# Snip:

