

G-Dec

OOP Assignment-2

21118

classmate

Date
Page

Problem Statement:

Develop a program in C++ to create a database of student's information system containing the following information: name, roll number, class, division, date of birth, blood group, contact address, telephone number, driving licence number & other. Construct the database with suitable member functions. Make use of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code & dynamic memory allocation operators - new & delete as well as exception handling.

Objectives

To learn concept of constructor, copy constructor, copy destructor, static member functions, friend class, this pointer, inline code & dynamic memory allocation operators - new & delete.

Outcomes:

Implement C++ personal information handling system using various OOP concepts.

Hardware requirements:

Manufacturer & model: Acer Swift-3 (Intel i5 8th gen)
Installed Memory: 8GB RAM, 512GB SSD.
Architecture 64-bit.

Software requirement:

Operating system: Windows-10 home Single language
IDE: code blocks version 20.03
Compiler: g++ (version 10.1.0)

Theory:

Constructor: A special method of class that will automatically

invoked when an instance of class is created.

- Types:
- Default constructor: no arguments & automatically called.
 - Parameterized constructor: some arguments
 - Copy constructor: object of another class as argument.

Destructor: Destructor is called when object is destroyed. Usually used to release allocated memory for that particular object.

Static Members: They have only single copy across all instances of a particular class.

Friends function: They can access private data members of the class.

Friend classes: They can access private data members of the class.

This pointer: pointer pointing to the calling object.

new & delete: These are the operators used to allocate memory dynamically on heap section & release that allocated memory after use.

Structure of classes

➤ class PersonalInfo &

private:

String blood_group, driving_licence_no;

String DOB, address;

int telephone_no;

public:

Record getData() & // get's data from user.

// code

{

inline void showData() & // fn shows data of instance

// code

{

friend class student


```

2> class Student {
private:
    static int cnt; // static variable to count all students
    string name, class;
    int roll-no, div;
    Personal_Info personal_info;
public:
    Student() { // constructor
        cnt++;
    }
    Student (const Student &s) { // copy constructor
        name = s.name
        // code
    }
    static int getStudentCnt() { // return total students in database
        // code
    }
    void getData() { // Method to get the data of student from user
        // code
    }
    inline void showData() { // Method to show data of student
        // code
    }
    ~Student() { // Destructor
        cnt--;
    }
}

```

Pseudo codes & use of concepts.

exception handling: It is used in the methods which takes data from the user.

consider the following code:

```
try {
    cout << "Enter roll no (b/w 1 to 86) : ";
    cin >> roll-no;
    if (roll-no < 1 || roll-no > 86)
        throw;
    // other code
} catch (int x) {
    if (x == 1)
        cout << "ERROR: Invalid roll no.\n";
    else if (x == 2)
        // code to handle other exceptions
} catch (...) {
    // default catch handler
}
```

new & delete:

These operators are used to create array of student data types.

Consider the following code:

Array creation: `Student* student-array = new Student[N];`

Array destruction: `delete[] student-array;`

Test cases

- 1) Entering student information : successfully done.
 - screenshot of the output is attached.
 - 2) Output student information: screenshot attached.
 - 3) Use of copy constructor
 - 4) Exception handling for invalid roll no
- } screenshots are attached.

Conclusion

Hence, we have successfully studied concept of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code & dynamic memory allocation operators new & delete.

Input and Output Demo:

Entering Student Information:

```
Enter data for student number : 1
Enter roll number (between 1 to 86) : 18
Enter name : Shubham Chemate
Enter class (class should be FE, SE, TE, BE) : SE
Enter div (div should be in 1 to 11) : 1
Enter Personal Info :
Enter DOB (DD/MM/YYYY) : 20/09/2020
Enter blood group : A+
Enter Address : xyz Road, Parner, A.Nagar - 414103
Enter Driving Licence Number : asdf1234
Enter Telephone Number : 1234567890
Student number 1 is created successfully.
```

Output (Student Information):

```
Student Info for roll number : 18
Name : Shubham Chemate
Class : SE, Div : 1
Date of Birth (DD/MM/YYYY) : 20/09/2020
Blood Group : A+
Address : xyz Road, Parner, A.Nagar - 414103
Driving Licence Number : asdf1234
Telephone Number : 1234567890
```

Output (Student Information – through copy constructor (Copies data of first student)):

```
=====Copy Constructor (Used for copying data of 1st Student)=====
Student Info for roll number : 18
Name : Shubham Chemate
Class : SE, Div : 1
Date of Birth (DD/MM/YYYY) : 20/09/2020
Blood Group : A+
Address : xyz Road, Parner, A.Nagar - 414103
Driving Licence Number : asdf1234
Telephone Number : 1234567890
```

Exception Handling (For invalid roll number):

```
For how many students do you want to create database?? 1
Enter data for student number : 1
Enter roll number (between 1 to 86) : 0
ERROR: Invalid Roll Number. It should be between (1 and 86)
```