#include<iostream>

using namespace std;

class student

{

    char name[10];

    int roll\_no;

    char branch[10];

    int maths, ld, ds, oops, co;

    float percentage;

    public:

        void getdata(int);

        void display();

        void calc\_pc();

};

void student :: getdata(int roll)

{

    roll\_no = roll;

    cout << "Enter name: ";

    cin >> name;

    cout << "Enter branch: ";

    cin >> branch;

    cout << "Enter marks in Maths, LD, DS, OOPS, CO respectively: ";

    cin >> maths >> ld >> ds >> oops >> co;

}

void student :: display()

{

    cout << "Name: " << name <<endl;

    cout << "Branch: " << branch << endl;

    cout << "Maths: " << maths << endl;

    cout << "LD: " << ld <<endl;

    cout << "DS: " << ds <<endl;

    cout << "OOPS: "<< oops <<endl;

    cout << "CO: " << co << endl;

}

void student :: calc\_pc()

{

    percentage = (maths + co + ld + oops + ds) / 5;

    cout << "Overall percentage: " << percentage <<endl;

}

int main()

{

    student s[3];

    int c, roll;

    cout << "Enter student details\n";

    for(int i = 0; i < 3; i++)

    {

        cout << "\nRoll number: " << i+1 <<endl;

        s[i].getdata(i+1);

    }

    do

    {

        cout << "Choose your option\n";

        cout << "1. Display student details\n";

        cout << "2. Calculate percentage and display\n";

        cout << "3. Exit\n";

        cin >> c;

        switch (c)

        {

            case 1:

                cout << "\nEnter roll number to display: ";

                cin >> roll;

                if (roll < 0 || roll > 3){

                    cout << "Invalid roll number\n";

                    break;

                }

                s[roll-1].display();

                break;

            case 2:

                cout << "\nEnter roll number to display: ";

                cin >> roll;

                if (roll < 0 || roll > 3){

                    cout << "Invalid roll number\n";

                    break;

                }

                s[roll-1].calc\_pc();

                break;

            case 3:

                break;

            default:

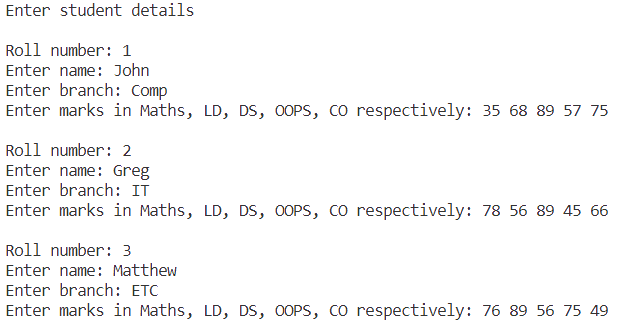
                cout << "Invalid input\n";

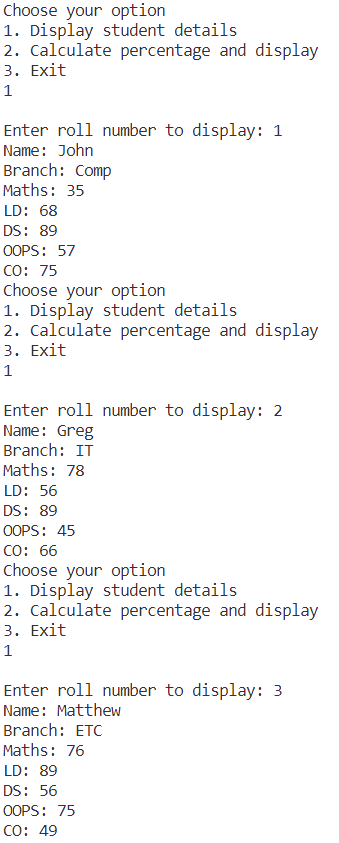
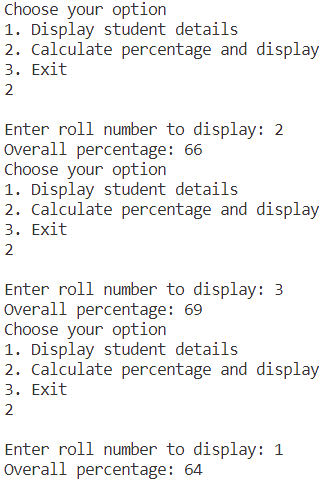
        }

    } while (c!=3);

}

**Output**





#include<iostream>

using namespace std;

class student

{

    char name[10];

    int roll\_no;

    char branch[10];

    int maths, ld, ds, oops, co;

    float percentage;

    public:

        void getdata(int);

        void display();

        void calc\_pc();

};

void student :: getdata(int roll)

{

    roll\_no = roll;

    cout << "Enter name: ";

    cin >> name;

    cout << "Enter branch: ";

    cin >> branch;

    cout << "Enter marks in Maths, LD, DS, OOPS, CO respectively: ";

    cin >> maths >> ld >> ds >> oops >> co;

}

void student :: display()

{

    cout << "Name: " << name <<endl;

    cout << "Branch: " << branch << endl;

    cout << "Maths: " << maths << endl;

    cout << "LD: " << ld <<endl;

    cout << "DS: " << ds <<endl;

    cout << "OOPS: "<< oops <<endl;

    cout << "CO: " << co << endl;

}

void student :: calc\_pc()

{

    percentage = (maths + co + ld + oops + ds) / 5;

    cout << "Overall percentage: " << percentage <<endl;

}

int main()

{

    student \*s;

    int n;

    cout << "Enter number of students";

    cin >> n;

    s = new student [n];

    int c, roll;

    cout << "Enter student details\n";

    for(int i = 0; i < n; i++)

    {

        cout << "\nRoll number: " << i+1 <<endl;

        s[i].getdata(i+1);

    }

    do

    {

        cout << "Choose your option\n";

        cout << "1. Display student details\n";

        cout << "2. Calculate percentage and display\n";

        cout << "3. Exit\n";

        cin >> c;

        switch (c)

        {

            case 1:

                cout << "\nEnter roll number to display: ";

                cin >> roll;

                if (roll < 0 || roll > n){

                    cout << "Invalid roll number\n";

                    break;

                }

                s[roll-1].display();

                break;

            case 2:

                cout << "\nEnter roll number to display: ";

                cin >> roll;

                if (roll < 0 || roll > n){

                    cout << "Invalid roll number\n";

                    break;

                }

                s[roll-1].calc\_pc();

                break;

            case 3:

                break;

            default:

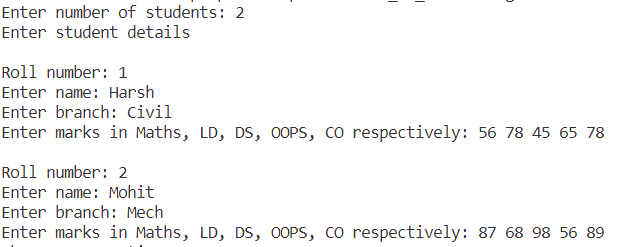
                cout << "Invalid input\n";

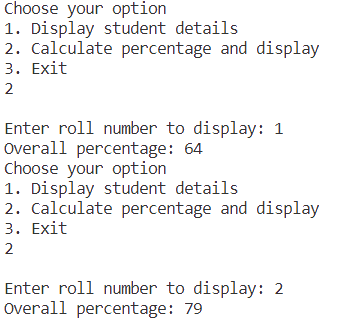
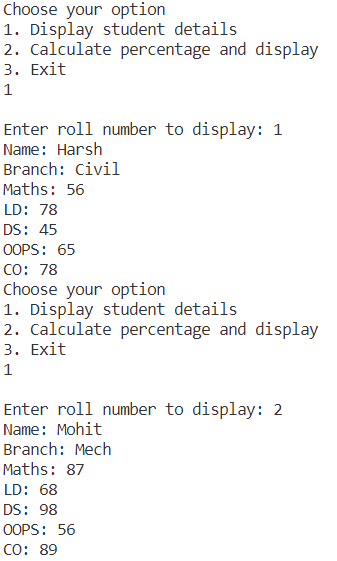
        }

    } while (c!=3);

}

**Output**





Experiment No: 1 Date: 11.08.2023

Aim: To study basics of C++ programming

|  |  |
| --- | --- |
| [A] | Write a C++ program to understand basic concepts of classes and objects (array of  objects). |
| [B] | Write a C++ program to understand concept of Dynamic Memory Allocation for the array  of objects. |

**Theory:**

C++ is a powerful and versatile programming language that combines the features of C with Object-Oriented Programming (OOP) concepts. It is widely used for developing a wide range of applications, from system software to game development. Understanding the basics of C++ is essential for anyone stepping into the world of programming.

**1. Structure of a C++ Program:** A C++ program consists of functions, and the primary function is the **main()** function. The program execution starts from **main()**.

**2. Variables and Data Types:** C++ supports various data types, including int, float, double, char, and more. Variables hold values of these data types and must be declared before use.

**3. Input and Output:** The **cin** and **cout** streams are used to take input from the user and display output, respectively.

**4. Operators:** C++ supports various operators like arithmetic, logical, relational, and assignment operators for performing different operations on variables.

**5. Control Structures:** C++ provides if-else, switch, while, for, and do-while loops for controlling the flow of execution.

**6. Functions:** Functions allow code reusability and modular programming. They are defined with a return type, name, and parameters.

**7. Arrays:** Arrays store multiple elements of the same data type. They are indexed starting from 0.

**8. Pointers:** Pointers hold memory addresses. They are used for dynamic memory allocation and direct memory manipulation.

**9. Classes and Objects:** C++ supports OOP concepts. Classes define blueprints for objects, while objects are instances of classes.

**10. Constructors and Destructors:** Constructors initialize objects, and destructors clean up resources when an object goes out of scope.

**11. Inheritance:** Inheritance allows a class to inherit properties and behaviors from another class. It supports the creation of hierarchical relationships.

**12. Polymorphism:** Polymorphism enables objects of different classes to be treated as objects of a common base class, simplifying code design.

**13. Encapsulation:** Encapsulation restricts access to certain parts of a class, promoting data hiding and security.

**14. Templates:** Templates allow you to create generic classes and functions that can work with various data types.

**15. Exception Handling:** Exception handling helps manage runtime errors, ensuring graceful program termination.

**16. Standard Template Library (STL):** The STL provides a collection of classes and functions for common data structures and algorithms.

**18. Memory Management:** C++ offers dynamic memory allocation through operators like **new** and **delete**.

**19. Preprocessor Directives:** Directives like **#include** and **#define** are used to include header files and define constants, respectively.