1. # include <iostream>
2. using namespace std;
3. int main()
4. {
5. char op;
6. float num1, num2;
7. cout << "Enter operator either + or - or \* or /: ";
8. cin >> op;
9. cout << "Enter two operands: ";
10. cin >> num1 >> num2;
11. switch(op)
12. {
13. case '+':
14. cout << num1+num2;
15. break;
16. case '-':
17. cout << num1-num2;
18. break;
19. case '\*':
20. cout << num1\*num2;
21. break;
22. case '/':
23. cout << num1/num2;
24. break;
25. default:
26. // If the operator is other than +, -, \* or /, error message is shown
27. cout << "Error! operator is not correct";
28. break;
29. }
30. return 0;
31. }

Procedural Programming

Function: Calculate numbers procedurally

Language: C++

// C++ program to demonstrate

// accessing of data members

#include <bits/stdc++.h>

using namespace std;

class Geeks

{

    // Access specifier

    public:

    // Data Members

    string geekname;

    // Member Functions()

    void printname()

    {

       cout << "Geekname is: " << geekname;

    }

};

int main() {

    // Declare an object of class geeks

    Geeks obj1;

    // accessing data member

    obj1.geekname = "Abhi";

    // accessing member function

    obj1.printname();

    return 0;

}

#include <bits/stdc++.h>

using namespace std;

class Geeks

{

    public:

    string geekname;

    int id;

    // printname is not defined inside class defination

    void printname();

    // printid is defined inside class defination

    void printid()

    {

        cout << "Geek id is: " << id;

    }

};

// Definition of printname using scope resolution operator ::

void Geeks::printname()

{

    cout << "Geekname is: " << geekname;

}

int main() {

    Geeks obj1;

    obj1.geekname = "xyz";

    obj1.id=15;

    // call printname()

    obj1.printname();

    cout << endl;

    // call printid()

    obj1.printid();

    return 0;

}

Object oriented Programming

Function: Creates “instances” of the same function as objects to calculate numbers

Language: C++

class Calculator { public: double Calculate(double x, char oper, double y); };

double Calculator::Calculate(double x, char oper, double y)

{ switch(oper) {

case '+': return x + y;

case '-': return x - y;

case '\*': return x \* y;

case '/': return x / y;

default: return 0; } }

Function programming

Function: calculates functions using only functions

Language: C++