**Programs:**

1. Overload the unary minus operator(-) to negate members of a class.

#include<iostream>

#include<conio.h>

using namespace std;

class temperatureLoc {

int tempDegree;

public:

//constructor

temperatureLoc() {

cout << "Enter the temperature of the location which is positive: ";

cin >> tempDegree;

}

//operator minus overloading

void operator -() {

tempDegree = -tempDegree;

}

void display() {

cout << "The temperature of the location is " << tempDegree << " degree." << endl;

return;

}

};

int main() {

temperatureLoc loc;

cout << "Positive Temperature: " << endl;

loc.display();

-loc;

cout << "Negative Temperature: " << endl;

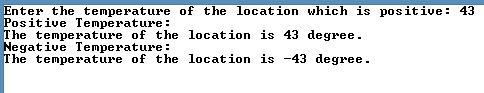
loc.display();

\_getch();

return 0;

}

**Output:**



1. Write a class Time having members hours and minutes. Overload unary increment operator to add a minute to minutes and check if minutes>=60 then increment hours by 1.

#include<iostream>

#include<conio.h>

using namespace std;

class Time {

int hours;

int minutes;

public:

//constructor

Time() {

cout << "Enter the intial value of the hours: ";

cin >> hours;

cout << "Enter the intial value of the minutes: ";

cin >> minutes;

}

//operator minus overloading

void operator ++() {

minutes = minutes + 1;

if (minutes >= 60) {

hours += 1;

minutes = 0;

}

}

//diplay current value of the members.

void display() {

cout <<"The time is " << hours<< ":"<< minutes<< endl;

return;

}

};

int main() {

Time obj;

for(int i = 0; i < 10; i++) {

++obj;

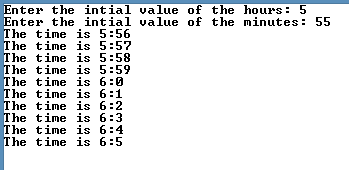
obj.display();

}

\_getch();

return 0;

}



**Output:**

1. Overload binary operators +, - , \* and / to add, subtract, multiply and divide two complex numbers. Let + and – be overloaded as a member functions and \* and / be overloaded as a friend function.

#include<iostream>

#include<conio.h>

using namespace std;

class complexNum {

int real, imag;

public:

complexNum(){

cout << "Enter the value of the real value: ";

cin >> real;

cout << "Enter the value of the imagnary: ";

cin >> imag;

}

complexNum(int a, int b) {

real = a;

imag = b;

}

void display() {

cout << "The result of the opeartion is " << real << " + " << imag << "i" << endl;

}

complexNum operator +(complexNum& obj) {

complexNum result(0,0);

result.real = real + obj.real;

result.imag = imag + obj.imag;

return result;

}

complexNum operator -(complexNum& obj) {

complexNum result(0, 0);

result.real = real - obj.real;

result.imag = imag - obj.imag;

return result;

}

friend complexNum operator \*(complexNum& obj1, complexNum& obj2);

friend complexNum operator /(complexNum& obj1, complexNum& obj2);

};

complexNum operator \*(complexNum& obj1, complexNum& obj2) {

complexNum result(0, 0);

result.real = (obj1.real \* obj2.real) - (obj1.imag \* obj2.imag);

result.imag = (obj1.real \* obj2.imag) + (obj1.imag \* obj2.real);

return result;

}

complexNum operator /(complexNum& obj1, complexNum& obj2) {

complexNum result(0, 0);

result.real = obj1.real + obj2.real;

result.imag = obj1.imag + obj2.imag;

return result;

}

int main() {

//object declarations

complexNum num1, num2, resultA(0,0), resultS(0,0), resultM(0,0), resultD(0,0);

resultA = num1 + num2;

cout << "\nADDITION" << endl;

resultA.display();

resultS = num1 - num2;

cout << "\nSUBTRATION" << endl;

resultS.display();

resultM = num1 \* num2;

cout << "\nMULTIPLICATION" << endl;

resultM.display();

resultD = num1 + num2;

cout << "\nDIVISION" << endl;

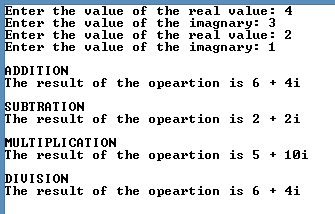
resultD.display();

\_getch();

return 0;

}

**Output:**



1. Write a program to overload the subscript operator [].

#include<iostream>

#include<conio.h>

using namespace std;

const int SIZE = 5;

class Subscript {

int arr[SIZE];

public:

Subscript() {

int i = 0;

cout << "Enter the values: " << endl;

while (i < SIZE) {

cout << "arr[" << i << "] = ";

cin >> arr[i];

i++;

}

}

int &operator [](int i) {

if (i > SIZE) {

cout << "Index out of bound" << endl;

return arr[0];

}

else {

return arr[i];

}

}

};

int main() {

Subscript obj;

cout << "--- DISPLAY ---" << endl;

cout << "arr[ 3 ] = ";

cout<<obj[3]<<endl;

cout << "arr[ 4 ] = ";

cout << obj[4] << endl;

cout << "arr[ 7 ] = ";

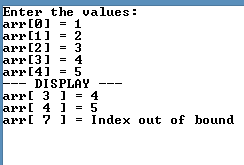
obj[7];

\_getch();

return 0;

}

**Output:**



1. Write a C++ Program to Compare Two Strings by Overloading == operator

#include<iostream>

#include<conio.h>

#include<string>

using namespace std;

class StringCom {

string str;

public:

//constructor

StringCom() {

cout << "Enter the string: ";

cin >> str;

}

//operator equal overloading

int operator ==(StringCom& obj) {

if (str == obj.str) {

return 1;

}

else {

return 0;

}

}

};

int main() {

StringCom str1, str2;

if (str1 == str2) {

cout << "The strings are equal" << endl;

}

else {

cout << "The strings are not equal" << endl;

}

\_getch();

return 0;

}

**Output:**

Not equal : Equal:

