**Program 01: Write a C++ program to get the size of different type of datatypes.**

**Code:**

#include <iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

cout << "Size of int: " << sizeof(int) << " bytes" << endl;

cout << "Size of char: " << sizeof(char) << " bytes" << endl;

cout << "Size of float: " << sizeof(float) << " bytes" << endl;

cout << "Size of double: " << sizeof(double) << " bytes" << endl;

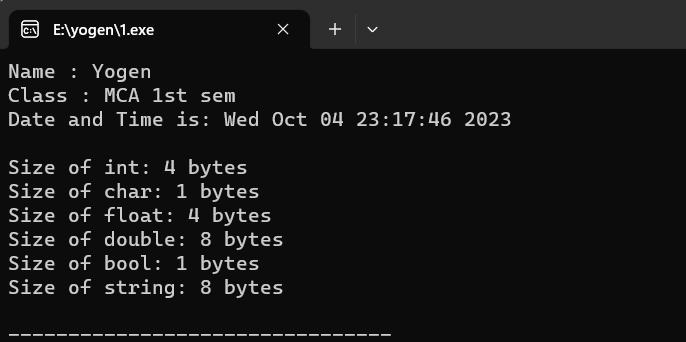
cout << "Size of bool: " << sizeof(bool) << " bytes" << endl;

cout << "Size of string: " << sizeof(str) << " bytes" << endl;

return 0;

}

**Output:**



**Program 02: Write a C++ program to perform arithmetic operations using arithmetic operators.**

**Code:**

#include<iostream>

#include<ctime>

using namespace std;

int main ()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int a, b;

cout << "Enter the value of a and value of b:";

cin >> a>>b;

int addition = a + b;

int difference = a - b;

int multiplication = a \* b;

float division = a / b;

int remainder = a % b;

cout << "Sum of "<<a<<" and "<<b<<"is = " << addition << endl;

cout << "Difference between " <<a<<" and "<<b<<"is = " << difference << endl;

cout << "Product of " <<a<<" and "<<b<<"is = " << multiplication << endl;

cout << "Division of " <<a<<" and "<<b<<"is = " << division << endl;

cout << "Remainder of " <<a<<" and "<<b<<"is = " << remainder << endl;

cout << "Pre Increment of ++a is = "<<++a<<endl;

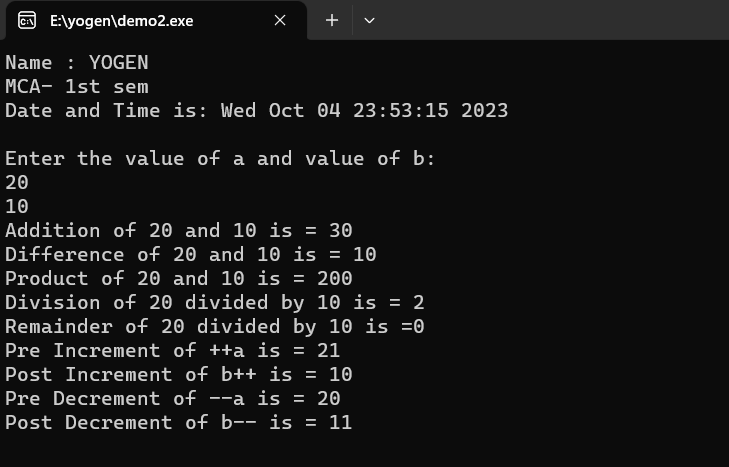
cout << "Post Increment of b++ is = "<<b++<<endl;

cout << "Pre Decrement of --a is = "<<--a<<endl;

cout << "Post Decrement of b-- is = "<<b--<<endl;

return 0;}

**Output :-**



**Program 03: Write a C++ program to demonstrate the different types of relational operators .**

**Code:**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int x,y;

cout<<"Enter first and second number "<<endl;

cin>>x>>y;

cout<<x<<" == "<<y<<" : "<<(x==y)<<endl;

cout<<x<<" != "<<y<<" : "<<(x!=y)<<endl;

cout<<x<<" <= "<<y<<" : "<<(x<=y)<<endl;

cout<<x<<" >= "<<y<<" : "<<(x>=y)<<endl;

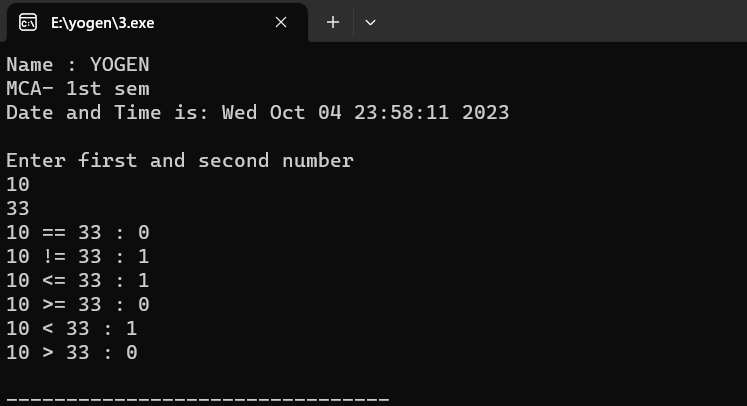
cout<<x<<" < "<<y<<" : "<<(x<y)<<endl;

cout<<x<<" > "<<y<<" : "<<(x>y)<<endl;

return 0;

}

**Output:**



**Program 04: Write a C++ program to demonstrate the logical operators (&&,||,!) .**

**Code:**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA - 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int a,b;

cout<<"Enter first binary number (0 or 1) :";

cin>>a;

cout<<"Enter second binary number (0 or 1) :";

cin>>b;

// Logical AND (&&) operator

cout << "Logical AND (&&) Operator:" << endl;

cout <<a<<" && "<<b<< " = "<<(a && b) << endl; // true && false = false

// Logical OR (||) operator

cout << "Logical OR (||) Operator:" << endl;

cout <<a<<" || "<<b<< " = "<<(a || b) << endl; // true || false = true

// Logical NOT (!) operator

cout << "Logical NOT (!) Operator:" << endl;

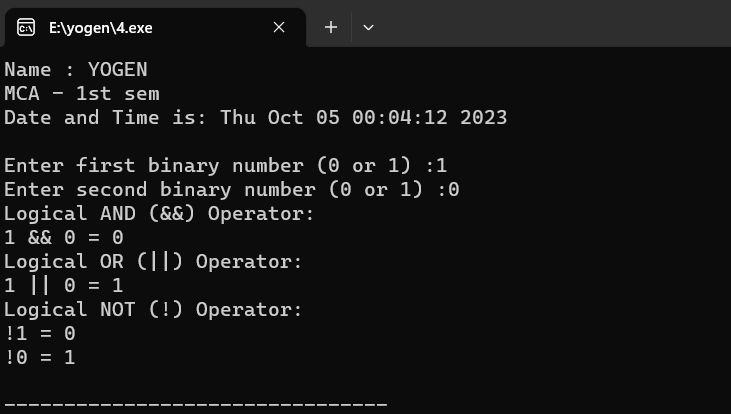
cout << "!"<<a <<" = " <<!a<< endl; // !true = false

cout << "!"<<b <<" = " <<!b<< endl; // !false = true

return 0;

}

**Output:-**



**Program 05: Write a C++ program to demonstrate the Bitwise operators .**

**Code :**

#include <iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int num1,num2;

cout<<"Enter a two number: "<<endl;

cin>>num1>>num2;

// Bitwise AND

int result\_and = num1 & num2;

cout << "Bitwise AND: " << result\_and <<endl;

// Bitwise OR

int result\_or = num1 | num2;

cout << "Bitwise OR: " << result\_or <<endl;

// Bitwise XOR

int result\_xor = num1 ^ num2;

cout << "Bitwise XOR: " << result\_xor <<endl;

// Bitwise Left Shift

int left\_shifted = num1 << 2;

cout << "Bitwise Left Shift: " << left\_shifted <<endl;

// Bitwise Right Shift

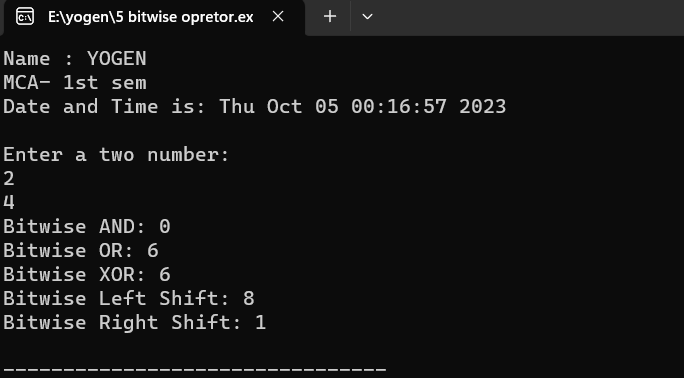
int right\_shifted = num1 >> 1;

cout << "Bitwise Right Shift: " << right\_shifted <<endl;

return 0;

}

**Output:**



**Program 06: Write a C++ program to read radius of a circle, calculate area and perimeter and display them.(using const constant).**

**Code:**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

const float PI = 3.14; //Using const keyword find Area and parameter of circle

int r;

float a,p;

cout<<"Enter a radius of circle:"<<endl;

cin>>r;

a= PI\*r\*r;

cout<<" The area of circle is: "<<a<<endl;

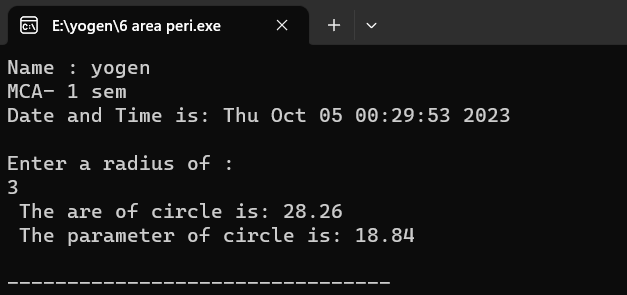
p=2\*PI\*r;

cout<<" The parameter of circle is: "<<p<<endl;

return 0;

}

**Output :-**



**Program 07: Write a C++ program for Assuming that res starts with the value 25 and p with 3,so print the following code:-**

**a) cout<<res--;**

**cout<<++res;**

**b) p=p\*++res;**

**Code :**

#include <iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA- 1ST sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int res = 25;

int p = 3;

// Part a)

cout << "a) ";

cout << res--; // Post-decrement res

cout << ++res; // Pre-increment res

cout << endl;

// Part b)

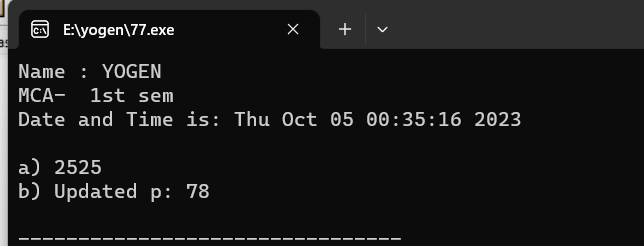
p = p \* ++res; // Pre-increment res and update p

cout << "b) Updated p: " << p <<endl;

return 0;

}

**Output:-**



**Program 08: Write a C++ program to input number of weeks day(1-7) and translate to its equivalent name of the day of the week using switch case.**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int dayNumber;

cout << "Enter a number between 1 to 7: ";

cin >> dayNumber;

switch (dayNumber)

{

case 1:

cout << "Monday" << endl;

break;

case 2:

cout << "Tuesday" << endl;

break;

case 3:

cout << "Wednesday" << endl;

break;

case 4:

cout << "Thrusday" << endl;

break;

case 5:

cout << "Friday" << endl;

break;

case 6:

cout << "Saturday" << endl;

break;

case 7:

cout << "Sunday" << endl;

break;

default:

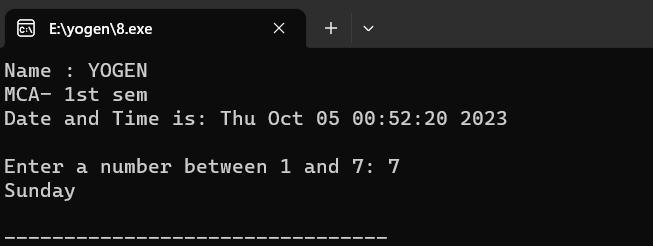
cout << "Invalid input! Please enter a number between 1 to 7." << endl;

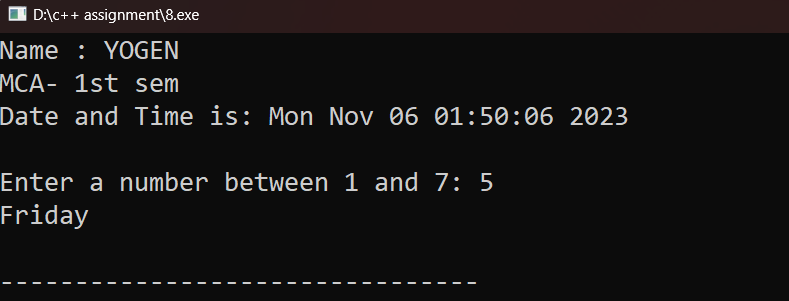
}

return 0;

}

**Output:**





**Program 09: Write a C++ program to make basic calculator using switch case.**

Code:

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

char operation;

double num1, num2;

cout << "Enter operator (+, -, \*, /): ";

cin >> operation;

cout << "Enter two numbers: ";

cin >> num1 >> num2;

switch (operation)

{

case '+':

cout << num1 << " + " << num2 << " = " << num1 + num2 << endl;

break;

case '-':

cout << num1 << " - " << num2 << " = " << num1 - num2 << endl;

break;

case '\*':

cout << num1 << " \* " << num2 << " = " << num1 \* num2 << endl;

break;

case '/':

cout << num1 << " / " << num2 << " = " << num1 / num2 << endl;

break;

default:

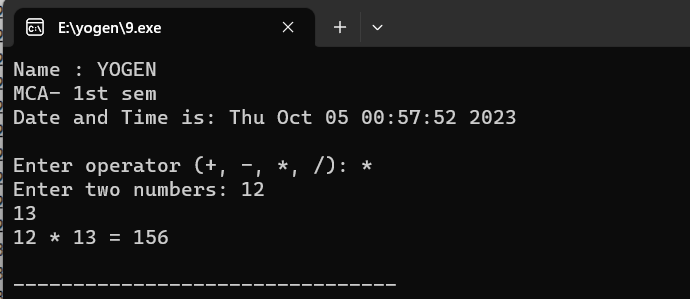
cout << "Invalid operator! Please enter +, -, \*, or /." << endl;

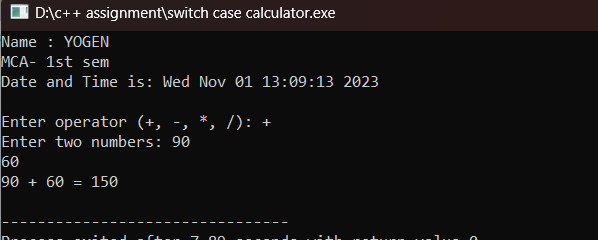
}

return 0;

}

**Output:**





**Program 10: Write a C++ program to find largest number among three number using ternary operator.**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int a, b, c, max;

cout << "Enter Three Integers\n";

cin >> a >> b >> c;

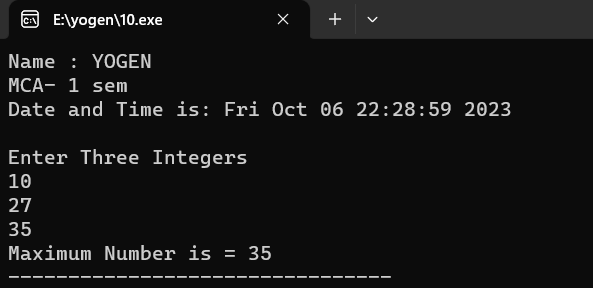
max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c); **//using ternary operator**

cout << "Maximum Number is = " << max;

return 0;

}

Output :



**Program11: Write a C++ program using for loop to print numbers from 1 to 10**.

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

for (int i = 1; i <= 10; i++){

cout << i << " ";

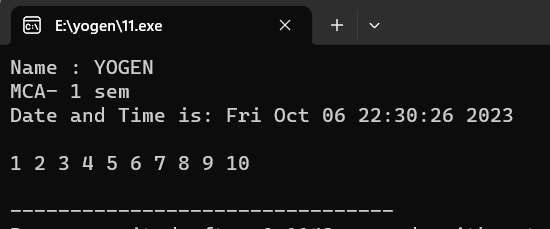
}

cout << endl;

return 0;

}

Output :

 **Program12: Write a C++ program to display 2,4,6,8…….,18,20 using while loop.**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int i=0;

while(i<=18) // using while loop

{

i+=2; // using assignment operator

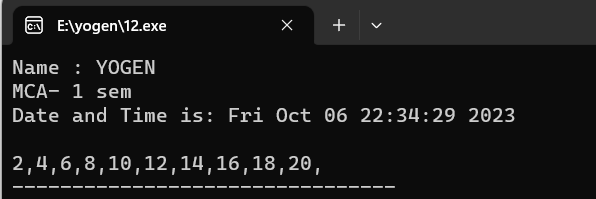
cout<<i<<",";

}

return 0;

}

Output :



**Program13: Write a c++ program to print following patterns :**

**A) \* \* \* \***

**\* \* \***

**\* \***

**\***

**B) \***

**\* \***

**\* \* \***

**\* \* \* \***

**C) 1**

**1 2**

**1 2 3**

**1 2 3 4**

**D) \***

**\* \***

**\* \* \***

**\* \* \* \***

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

void details()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

}

void triangle1()

{

int n;

cout<<"Enter a number : ";

cin>>n;

cout<<"A)"<<endl;

for(int i=n; i>=1; i--)

{

for(int j=1; j<=i;j++)

{

cout<<"\* ";

}

cout<<endl;

}

}

void triangle2()

{

int n;

cout<<"Enter a number : ";

cin>>n;

cout<<"B)"<<endl;

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= i; j++)

{

cout <<j<<" ";

}

cout <<endl;

}

}

void triangle3()

{

int n;

cout<<"Enter a number : ";

cin>>n;

cout<<"C)"<<endl;

for(int i=1;i<=4;i++)

{

for(int s=1;s<=i;s++)

{

cout<<"\* ";

}

cout<<endl;

}

}

void triangle4()

{

int n;

cout<<"Enter a number : ";

cin>>n;

cout<<"D)"<<endl;

for (int i = n; i > 0; i--)

{

for (int j = 1; j <= n; j++)

{

if (j >= i)

{

cout << "\* ";

}

else

{

cout << " ";

}

}

cout << endl;

}

}

int main()

{

details();

triangle1();

triangle2();

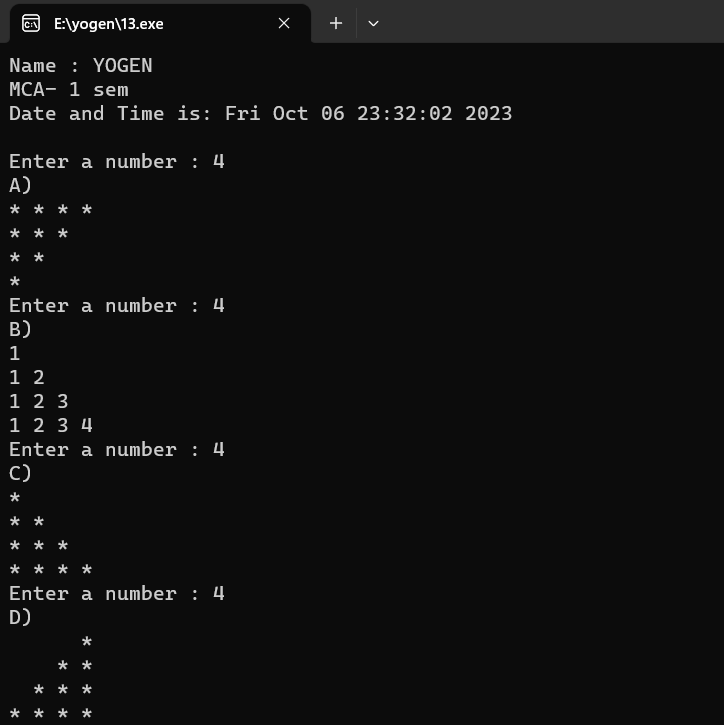
triangle3();

triangle4();

return 0;

}

Output:



**Program14: Write a C++ program to display the cube of the number upto an integer.**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

int main(){

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int range;

cout<<"Enter range : ";

cin>>range;

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

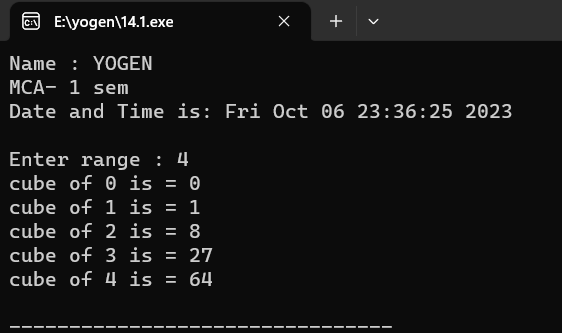
cout<<"cube of "<<i<<" is = "<<i\*i\*i<<endl;

}

return 0;

}

Output:



**Program15: Write a C++ program to check for equality of two numbers without using arithmetic or comparison operator.**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

//using binary repratation for equal check

bool areEqual(int num1, int num2)

{

return !(num1 ^ num2);

}

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int num1, num2;

cout << "Enter the first number: ";

cin >> num1;

cout << "Enter the second number: ";

cin >> num2;

if (areEqual(num1, num2))

{

cout << "The numbers are equal." << endl;

}

else

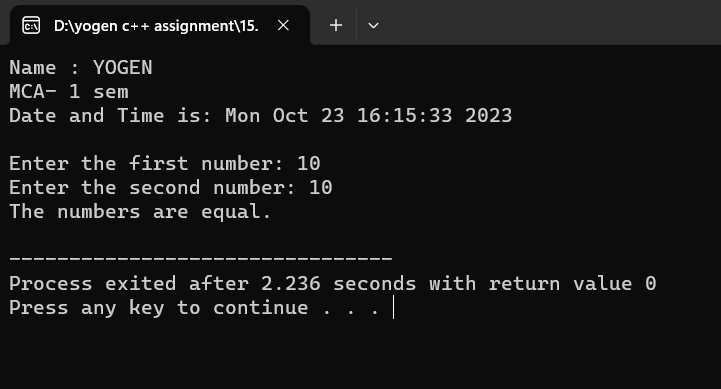
{

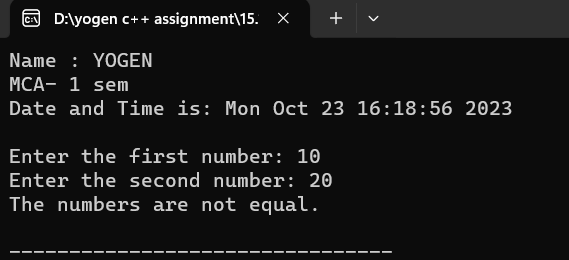
cout << "The numbers are not equal." << endl;

}

return 0;

Output:



**Program 16: Write a C++ program to calculate value of 132 x 8 without using “ \* ”operator**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int result;

int a=132;

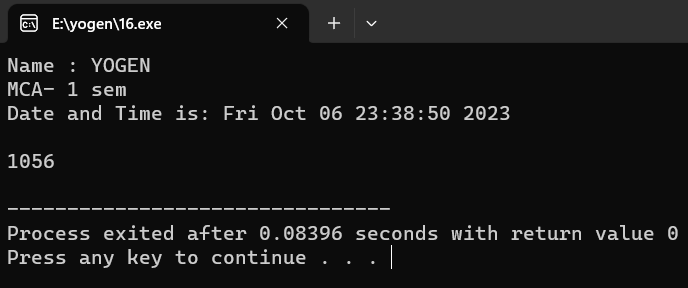
result=a<<3;

cout<<result<<endl;

return 0;

}

Output:



**Program 17: Write a C++ program to find Area of rectangle (using #define).**

**Code :**

#include <iostream>

#include<ctime>

using namespace std;

#define AREA(length, width) (length \* width) //declaration of #define

int main(){

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

double length, width;

cout << "Enter the length of the rectangle: ";

cin >> length;

cout << "Enter the width of the rectangle: ";

cin >> width;

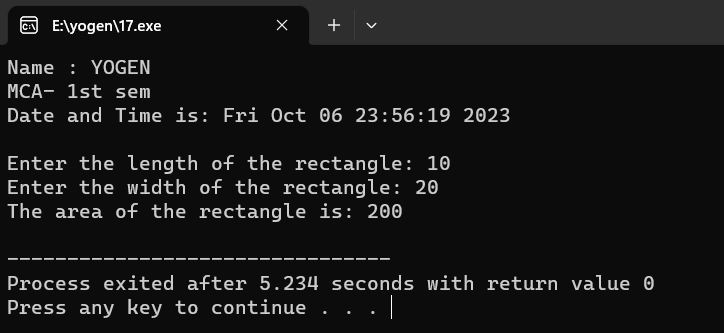
double area = AREA(length, width); // Calculate the area using the macro

cout << "The area of the rectangle is: " << area << endl;

return 0;

}

Output :



**Program 18: Write a C++ program to demonstrate explicitly typecasting..**

**Code :**

#include <iostream>

#include<ctime>

using namespace std;

int main() {

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

double s = 3.14159;

int r;

// Explicitly typecast a double to an int

r = (int)s;

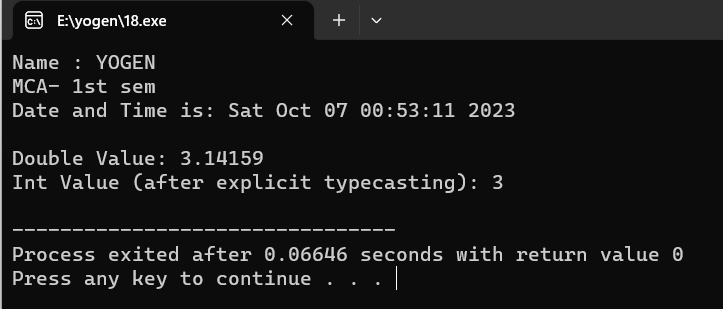
cout << "Double Value: " << s << endl;

cout << "Int Value (after explicit typecasting): " << r << endl;

return 0;

}

Output:



**Program 19: Write a C++ program to display addition of first 1 to 20 odd numbers and also display addition of first 1 to 20 even number**

**Code :**

#include<iostream>

#include<ctime>

using namespace std;

void details()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

}

void EvenOdd()

{

int Even=0;

int Odd=0;

for(int i=0;i<=40;i++)

{

if(i%2==0) {

Even+=i;

}

else {

Odd+=i;

}

}

cout<<"Total of first 20 even number is = "<<Even<<endl;

cout<<"Total of first 20 odd number is = "<<Odd;

}

int main(){

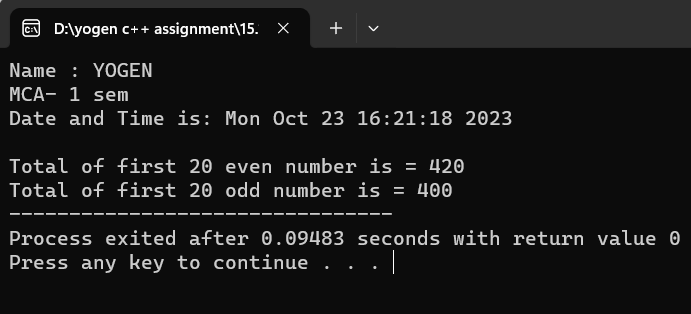
details();

EvenOdd();

return 0;

}

Output:



**Program 20: Write a C++ program for a given problem – where age will be taken as input by the user and if age is greater than 18 and gender is “Male(M/m)” than print message to send him to “Room number 10” for voting and if gender is “Female(F/f)” than print message to send her to “Room number 12”,if gender is none of these two than send them to “Room number 8”.also given message “Not eligible for voting” in case of age is less than 18.(using nested if control structure)**

**Code:**

#include <iostream>

using namespace std;

int main() {

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 29-09-2023 "<<endl<<endl;

int age;

char gender;

cout << "Enter your age: ";

cin >> age;

cout << "Enter your gender (M/m for Male, F/f for Female): ";

cin >> gender;

if (age >= 18) {

if (gender == 'M' || gender == 'm') {

cout << "Send him to Room number 10 for voting." << endl;

}

else if (gender == 'F' || gender == 'f') {

cout << "Send her to Room number 12 for voting." << endl;

}

else {

cout << "Gender not recognized. Send to Room number 8." << endl;

}

}

else {

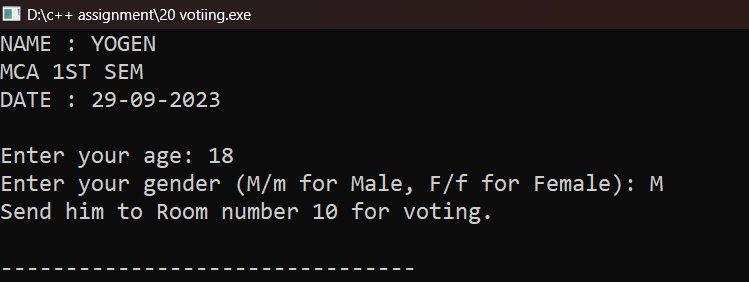
cout << "Not eligible for voting." << endl;

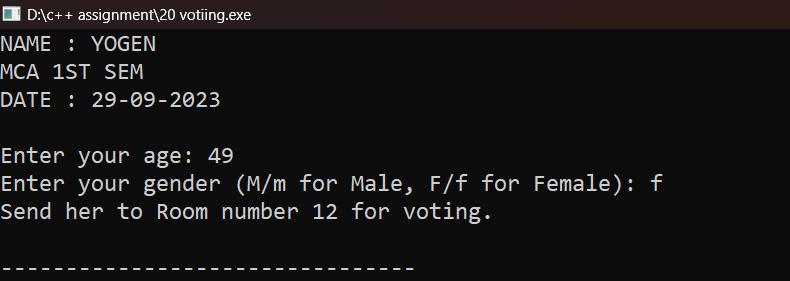
}

return 0;

}

Output:







**Program 21: Write a C++ program to print Hollow square pattern using for loop..**

**Code:**

#include <iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"YOGEN "<<endl;

cout<<"MCA- 1st sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int n;

cout << "Enter the number of rows or columns: ";

cin >> n;

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= n; j++)

{

// Print '\*' for the outer row, inner row, outer column, and inner column

if (i == 1 || i == n || j == 1 || j == n)

{

cout << "\* ";

}

else

{

cout << " "; // Print a space for the interior of the square

}

}

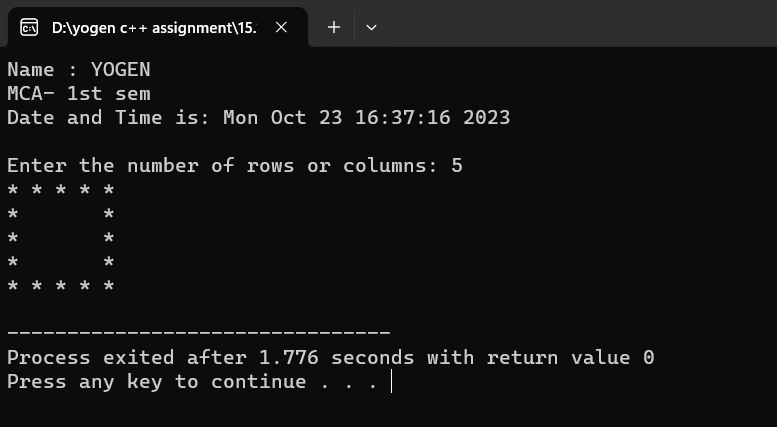
cout << endl;

}

return 0;

}

Output :



**Program22: Write a C++ program to take input between 1-25 at runtime and display “Thank you user” for selecting number between 1-25 otherwise display ”please enter number between 1-25 only!!”.(using while loop).**

**Code:**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int number;

cout<<"Enter number : ";

cin>>number;

while(number<=25)

{

cout<<" Thank you ";

return 0;

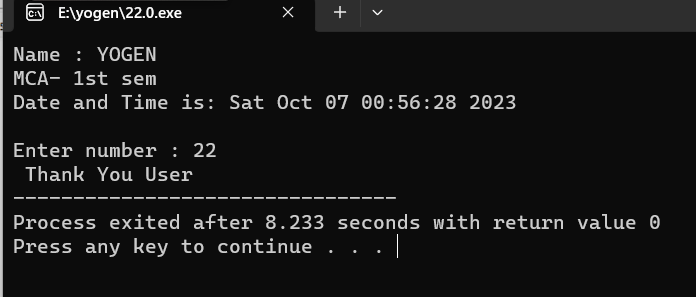
}

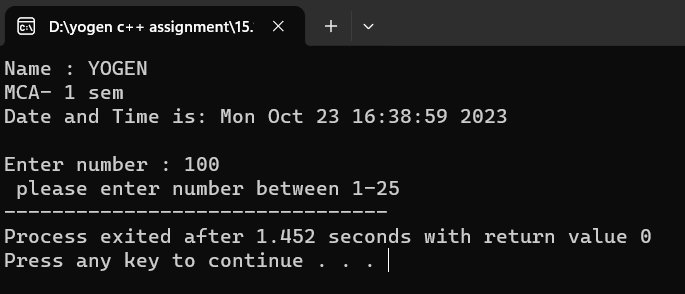
cout<<" please enter number between 1-25 ";

return 0

}

**Output:**



****

**Program23: Write a C++ program to find number is positive, negative or zero using “ goto ” jump statement..**

**Code:**

#include <iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

int number;

cout << "Enter a number: ";

cin >> number;

if (number > 0)

{

goto positive;

}

else if (number < 0)

{

goto negative;

}

else

{

goto zero;

}

positive:

cout << "The number is positive." << endl;

goto end;

negative:

cout << "The number is negative." << endl;

goto end;

zero:

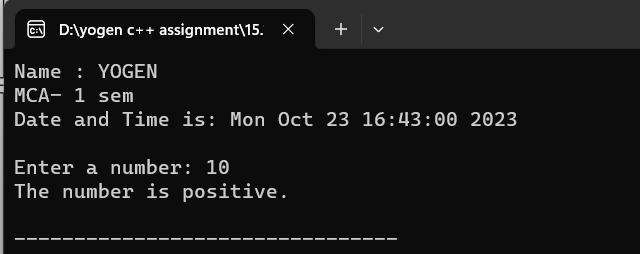
cout << "The number is zero." << endl;

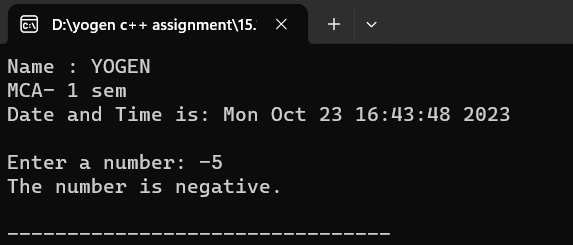
end:

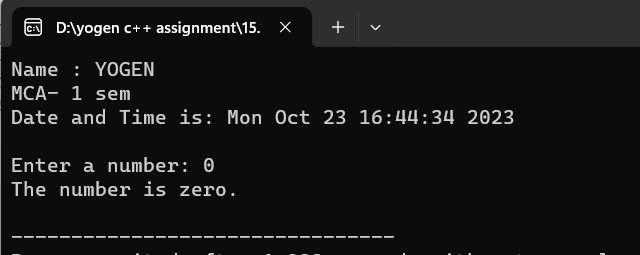
return **0;**

**}**

**Output:**

****





**Program24: Write a C++ program to differentiate break and continue jump statement.**

**Code:**

#include<iostream>

#include<ctime>

using namespace std;

int main()

{

cout<<"Name : YOGEN "<<endl;

cout<<"MCA- 1 sem"<<endl;

time\_t now = time(0);

char\* dt = ctime(&now);

cout<<"Date and Time is: "<<dt<<endl;

for (int i = 0; i < 15; i++)

{

if (i == 5)

{

break; // This will exit the loop completely.

}

cout << i << " ";

}

cout << endl;

for (int i = 0; i < 15; i++)

{

if (i == 5)

{

continue; // This will skip to the next iteration of the loop.

}

cout << i << " ";

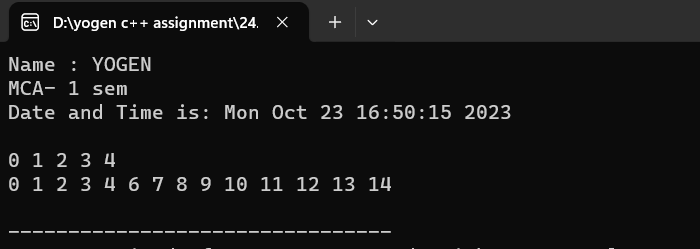
}

cout << endl;

return 0;

}

Output:



**Program25: write a C++ program to display detail of 5 student detail should contained student name, roll number , marks using structure.**

**Code:**

#include<iostream>

using namespace std;

struct Student{

string name;

int roll\_no;

int marks;

};

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

Student s[5];

for(int i=0;i<5;i++){

cout<<"Enter the details for student "<<i+1<<endl;

cout<<"Name : ";

cin>>s[i].name;

cout<<"Roll no : ";

cin>>s[i].roll\_no;

cout<<"Marks : ";

cin>>s[i].marks;

cout<<endl;

}

cout << "Student Details:" << endl;

for (int i = 0; i < 5; i++)

{

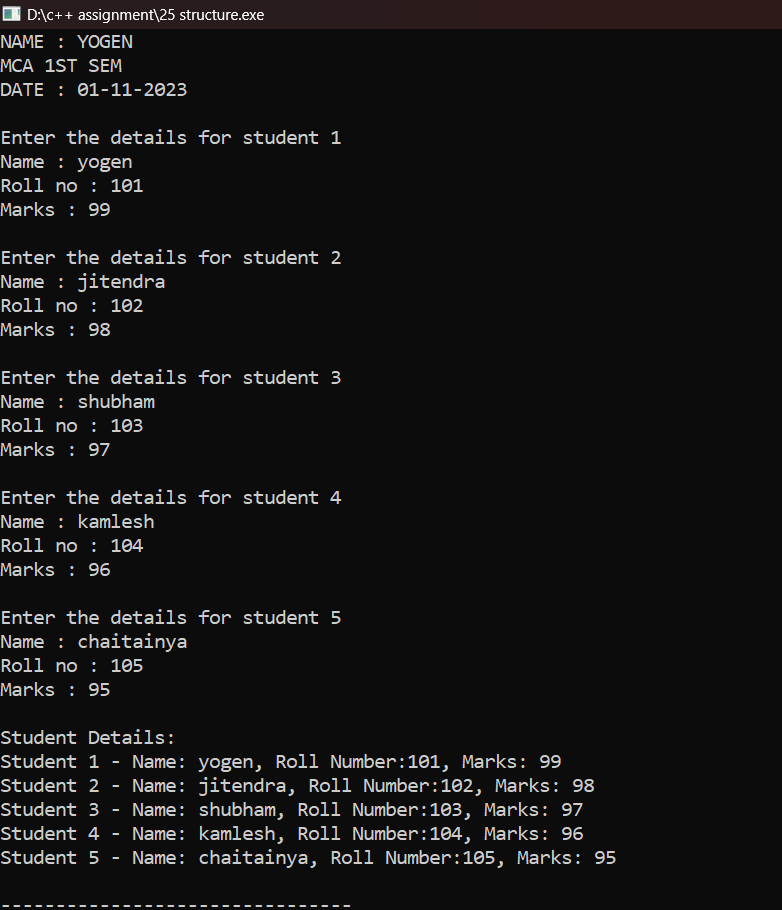
cout << "Student " << i + 1 << " - Name: "<<s[i].name << ", Roll Number:"<<s[i].roll\_no << ", Marks: " << s[i].marks << endl;

}

return 0;

}

**Output**

****

**Program26: Write a C++ program to create a structure named “Date” which contains three member Day, month, year and display current date entering by the user using function definition.**

**Code:**

#include<iostream>

using namespace std;

struct Date{

int day,month,year;

}s;

void show(){ // function definition

cout<<"Enter Day "<<endl;

cin>>s.day;

cout<<"Enter Month "<<endl;

cin>>s.month;

cout<<"Enter Year "<<endl;

cin>>s.year;

cout<<"Date : "<<s.day<<"/"<<s.month<<"/"<<s.year;

}

int main(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

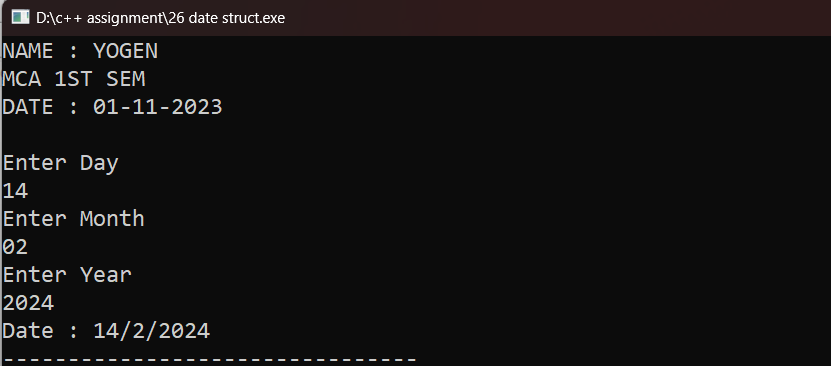
cout<<"DATE : 01-11-2023 "<<endl<<endl;

show();

return 0

}

**Output:**



**Program27: Write a C++ program to demonstrate enum with switch case.**

**Code:**

#include<iostream>

using namespace std;

enum Days{SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY};

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

Days today=FRIDAY;

switch (today) {

case SUNDAY:

cout << "It's Sunday. It's the weekend!" << endl;

break;

case MONDAY:

cout << "It's Monday. The workweek begins." << endl;

break;

case TUESDAY:

cout << "It's Tuesday. Work continues." << endl;

break;

case WEDNESDAY:

cout << "It's Wednesday. Midweek point." << endl;

break;

case THURSDAY:

cout << "It's Thursday. Almost there!" << endl;

break;

case FRIDAY:

cout << "It's Friday. The weekend is near." << endl;

break;

case SATURDAY:

cout << "It's Saturday. Enjoy the weekend!" << endl;

break;

default:

cout << "Invalid day." << endl;

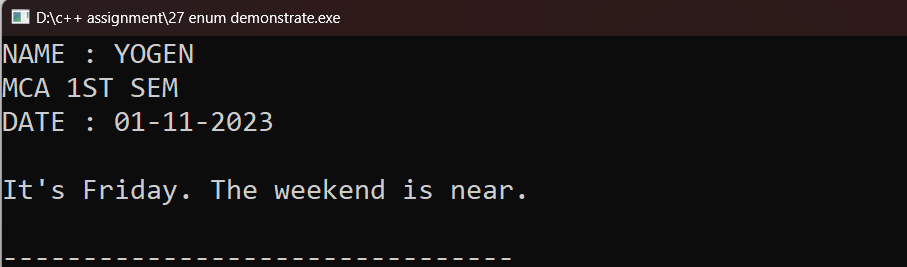
break;

}

return 0;

}

**Output:**

****

**Program 28. Write a C++ program to create an enum having number of enum list or elements and count the size of elements inside the enum.**

**Code:**

#include<iostream>

using namespace std;

enum count{sun,mon,tue,wed,thr,fri,sat}; // declare enum with elements

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int ctr=0;

for(int i=sun;i<=sat;i++){

ctr++ ;

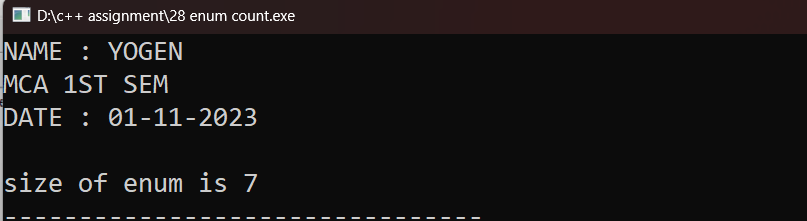
}

cout<<"size of enum is "<<ctr;

return 0;

}

Output :



**Program 29. Write a C++ program to swap two values without using third variable.**

**Code:**

#include<iostream>

using namespace std;

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int a, b;

cout << "Enter two values to swap:" << endl;

cout << "Enter the first value : ";

cin >> a;

cout << "Enter the second value : ";

cin >> b;

cout << "Before swapping: a = " << a << ", b = " << b << endl; // Swap the values

a = a \* b;

b = a / b;

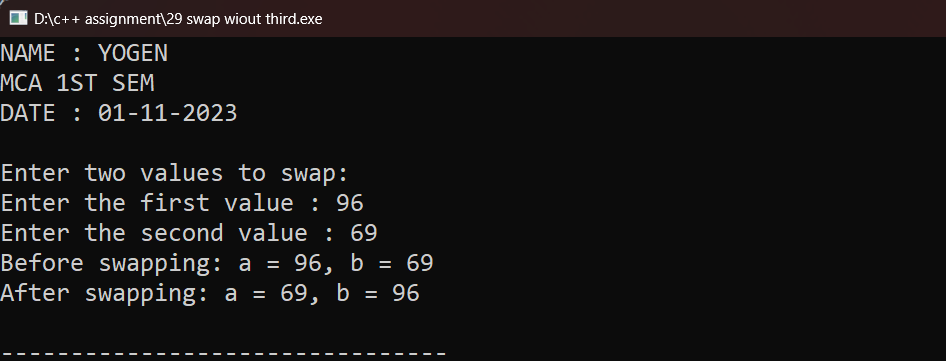
a = a / b;

cout << "After swapping: a = " << a << ", b = " << b << endl;

return 0;

}

**Output:**



**Program 30. Write a C++ program to swap two values using third variable of call by address function invoking.**

**Code:**

#include<iostream>

using namespace std;

void swap(int \*x,int\*y){

int temp =\*x;

\*x=\*y;

\*y=temp;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int a,b;

cout<<"Enter the first value for swap ";

cin>>a;

cout<<"Enter the second value for swap ";

cin>>b;

cout<<"Before swap value of a is = "<<a<<" and b is = "<<b<<endl;

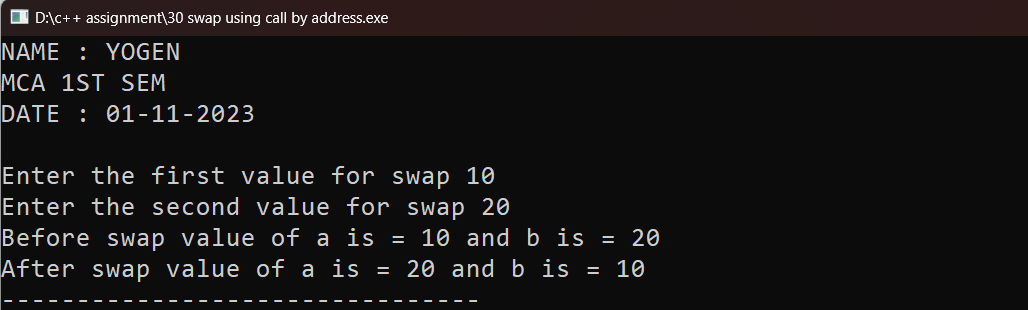
swap(a,b); // call by address

cout<<"After swap value of a is = "<<a<<" and b is = "<<b;

return 0;

}

**Output:**

****

**Program 31. Write a C++ program to illustrate working of call by value of a function invoking.**

**Code:**

#include<iostream>

using namespace std;

void sum(int x, int y){

int z=x+y;

cout<<"Sum is = "<<z;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int a,b;

cout<<"Enter first number = ";

cin>>a;

cout<<"Enter second number = ";

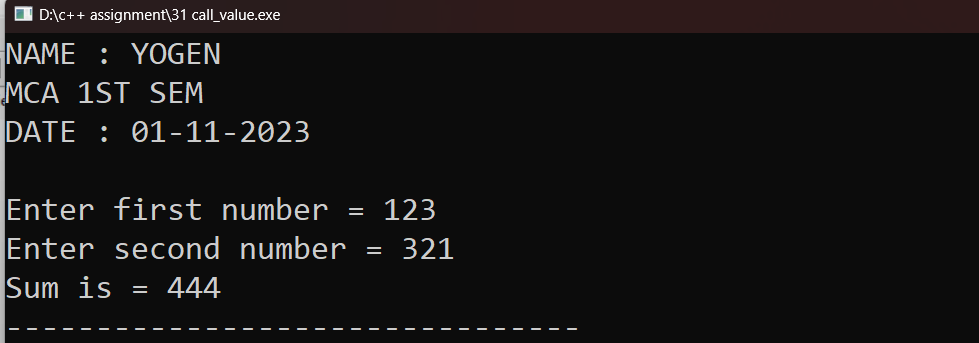
cin>>b;

sum(a,b); // function call by value

return 0;

}

Output:



**Program 32. Write a C++ program to illustrate working of call by reference method of a function invoking.**

**Code:**

#include<iostream>

using namespace std;

void sum(int &x, int &y){ // copy the address of actual parameter into formal parameter

int z=x+y;

cout<<"Sum is = "<<z;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int a,b;

cout<<"Enter first number = ";

cin>>a;

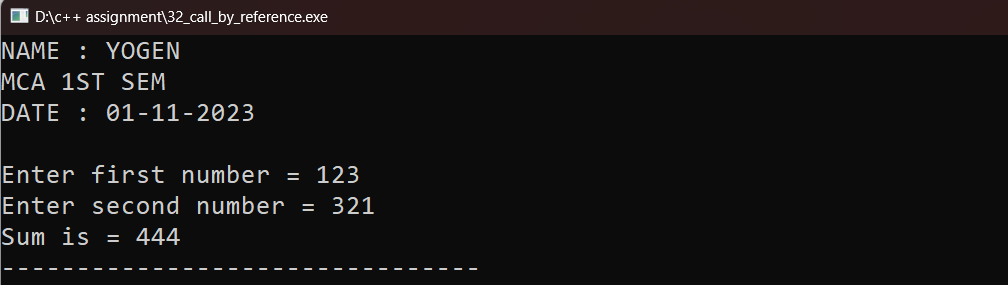
cout<<"Enter second number = ";

cin>>b;

sum(a,b); // passing actual parameter

return 0;

}Output:



**Program 33. Write a C++ program to calculate simple interest using default arguments.**

**Code:**

#include<iostream>

using namespace std;

double interest(double P , float R = 5.0, float T = 3.0) {

return (P\*R\*T)/100.0;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

double P;

cout<<"Enter a Principal amount: "<<endl;

cin>>P;

cout<<"simple interest is = "<<interest(P);

return 0;

}

Output:



**Program 34. Write a C++ program using function template to add two integers and two float number.**

**Code:**

#include<iostream>

using namespace std;

template <class T,class C>

T Addition (T x, T y, C a, C b)

{

cout<<"Sum of int is = "<<x+y<<endl;

cout<<"Sum of float is = "<<a+b<<endl;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int x,y;

float a,b;

cout<<"Enter two integer no. "<<endl;

cin>>x>>y;

cout<<"Enter two float no. "<<endl;

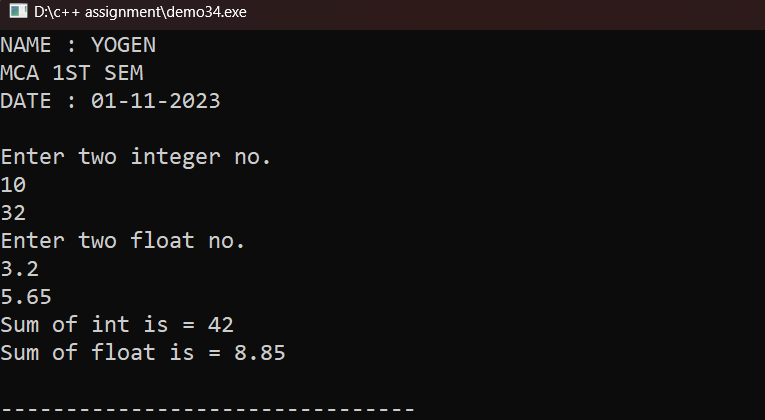
cin>>a>>b;

Addition(x,y,a,b);

return 0;

}

Output:



**Program 35. Write a C++ program to create simple calculator using class templates.**

**Code:**

#include<iostream>

using namespace std;

template <class T>

class Calculator {

T num1;

T num2;

public:

Calculator(T x, T y) : num1(x), num2(y) {}

T add() {

return num1 + num2;

}

T subtract() {

return num1 - num2;

}

T multiply() {

return num1 \* num2;

}

T divide() {

if (num2 == 0) {

cout<< "Error: Division by zero." << endl;

return 0;

}

return num1 / num2;

}

};

int main() {

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl;

double num1, num2;

char operation;

cout << "Choose an operation (+, -, \*, /): ";

cin >> operation;

cout << "Enter two numbers: "<<endl;

cin >> num1 >> num2;

Calculator<double> calculator(num1, num2);

switch (operation) {

case '+':

cout << "sum is = : " << calculator.add() << endl;

break;

case '-':

cout << "subtraction is = : " << calculator.subtract() << endl;

break;

case '\*':

cout << "multiplication is = : " << calculator.multiply() << endl;

break;

case '/':

cout << "divison is = : " << calculator.divide() << endl;

break;

default:

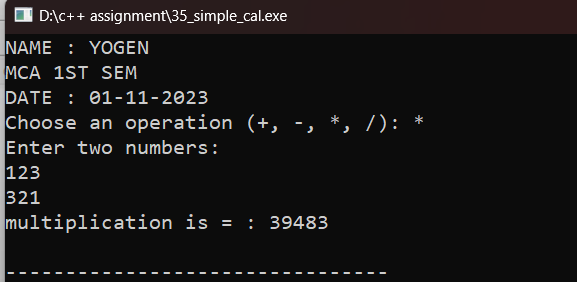
cout<< "Invalid operation." << endl;

break;

}

return 0;

**Output:**



**Program 36. Write a C++ program using inline function to calculate area of circle.**

**Code:**

#include<iostream>

using namespace std;

inline float AreaCircle(float r,float pie=3.14){ // Using inline function

return pie\*r\*r;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

float r;

cout<<"Enter the radius of circle is = ";

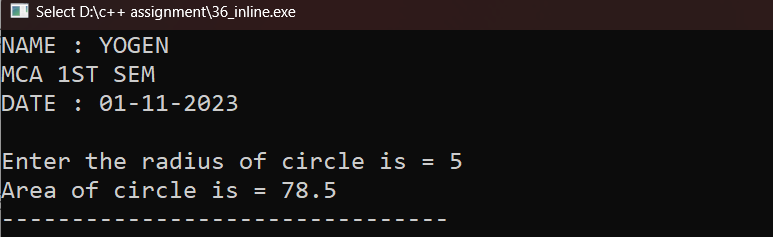
cin>>r;

cout<<"Area of circle is = " <<AreaCircle(r) ;

return 0;

}

**Output:**



**Program 37. Write a C++ program to demonstrate function overloading.**

**Code:**

#include<iostream>

using namespace std;

int add(int a, int b){

return a+b;

}

double add(double a,double b){

return a+b;

}

string add(string x, string y){

return x+y;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int x=5,y=10;

double c=5.857,d=8.766;

string r="YOGEN ",s="CHANDRAKAR";

int result1 = add(x,y);

double result2 = add(c,d);

string result3 = add(r,s);

cout<<"Addition of integer value : "<<result1<<endl;

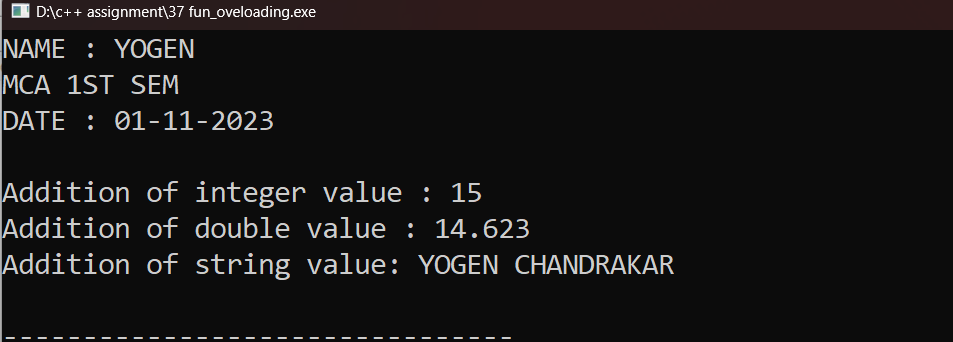
cout<<"Addition of double value : "<<result2<<endl;

cout<<"Addition of string value: "<<result3<<endl;

return 0;

}

Output :



**Program 38.write a C++ program to find the size of 1-D, 2-D, & multidimensional array.**

**Code:**

#include<iostream>

using namespace std;

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

//size of one D array

int oneDarry[] ={1,2,3,4,5,6};

int size1D = sizeof(oneDarry)/sizeof(oneDarry[0]);

cout<<"Size of 1D array is : "<<size1D<<endl;

//size of 2D array

int twoDarry[3][4] ={{1,2,3,4},{5,6,7,8},{9,10,11,12}};

int row2D =sizeof(twoDarry)/sizeof(twoDarry[0]);

int col2D =sizeof(twoDarry[0])/sizeof(twoDarry[0][0]);

cout<<"Size of 2D array is : "<<row2D<<"rows \*"<<col2D<<"columns"<<endl;

//size of 3D(multi dimension) array

int threeDarry[2][3][2] ={{

{1,2},{3,4},{5,6}},

{{7,8},{9,10},{11,12}}

};

int dim1 = sizeof(threeDarry)/sizeof(threeDarry[0]);

int dim2 = sizeof(threeDarry[0])/sizeof(threeDarry[0][0]);

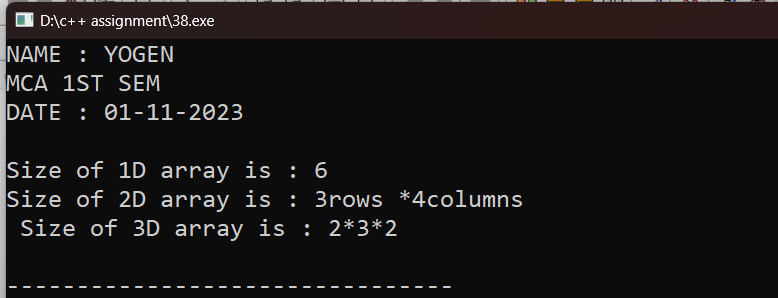
int dim3 = sizeof(threeDarry[0][0])/sizeof(threeDarry[0][0][0]);

cout<<" Size of 3D array is : "<<dim1<<"\*"<<dim2<<"\*"<<dim3<<endl;

return 0;

}

Output:-



**Program 39. write a C++ program create and display one-D array of size 7 and also display average of all the element.**

**Code:**

#include<iostream>

using namespace std;

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int oneDarry[7];

//for enter values

for(int i=0;i<7;i++){

cout<<"Enter element "<<i+1<<":";

cin>>oneDarry[i];

}

//calculate sum

int sum=0;

for(int i=0; i<7;i++){

sum += oneDarry[i];

int avg =sum/7;

}

cout<<"element is: ";

for(int i=0;i<7;i++){

cout<<oneDarry[i]<<" ";

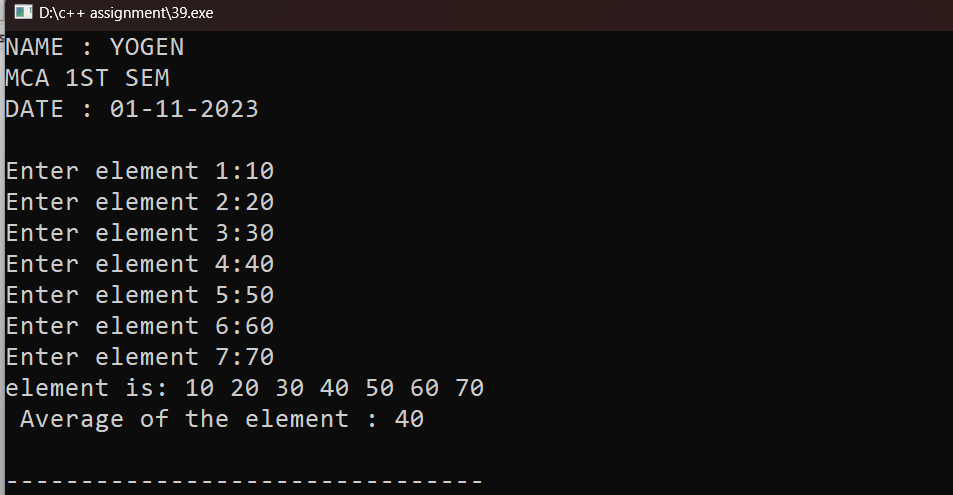
}

cout<<"\n Average of the element : "<<avg<<endl;

return 0;

}

Output;



**Program 40. Write a C++ program to input 5 number in an array and print all the number from the backside of the array.**

**Code:**

#include<iostream>

using namespace std;

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

int size;

cout<<"enter the size of array is = ";

cin>>size;

int number[size];

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

cout<<"Enter number "<<i+1<<":";

cin>>number[i];

}

cout<<"number in reverse order: "; ////printing number from backside

for(int i= size-1;i>=0; i--){

cout<<number[i]<<",";

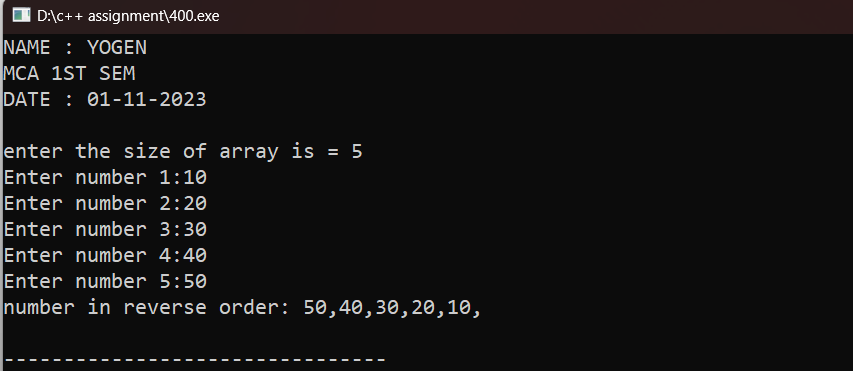
}

cout<<endl;

return 0 ;

}

Output:



**Program 41. Write a C++ program create class named “Student”, having two data member of private specifier name rollno and marks. and make marks data member as array of size 5. Student class also contains public member function named getdata( ) , showdata( ), and totalmarks( ) which will define outside of the class. getdata( ) will take input from the user only, showdata( ) will show the input data from the user, and totalmarks( ) will sum all the 5 marks of subject and display the total marks.**

**Code:**

#include<iostream>

using namespace std;

class student

{

int roll\_no;

int marks[5];

public:

void getdata();

void showdata();

void totalmarks();

};

void student :: getdata()

{

cout<<"enter the roll no"<<endl;

cin>>roll\_no;

cout<<"enter 5 marks "<<endl;

for (int i=0;i<5;i++)

{

cin>>marks[i];

}

}

void student :: showdata()

{ int i,sum;

cout<<"roll no is ="<<roll\_no<<endl;

// cout<<" marks is"<<marks[i];

cout<<"marks is"<<endl;

for (int i=0;i<5;i++)

{

cout<<marks[i]<<endl;

}

cout<<"total marks is ";

for (int i=0;i<5;i++)

{

sum += marks[i];

}

cout<<sum;

}

void student :: totalmarks()

{

int i, sum;

for (int i=0;i<5;i++)

{

// cin>>marks[i];

sum += marks[i];

}

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

student s;

s.getdata();

s.showdata();

s.totalmarks();

return 0;

}

Output:-



**Program 42. Write a C++ program to find greatest number among three numbers implementing the nesting of member function.**

**Code:**

#include<iostream>

using namespace std;

class ThreeNum{

public :

int a , b ,c ;

int max(int x, int y){

return x > y ? x : y ;

}

void display(){

cout << "a : " <<a ;

cout << ", b : " <<b ;

cout << ", c : " <<c << endl ;

}

void displayMax(){

cout << "max : " << max(a,max(b,c)) << endl ;

}

};

int main(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 04-11-2023 "<<endl<<endl;

ThreeNum t ;

t.a = 200 ;

t.b = 305 ;

t.c = 150 ;

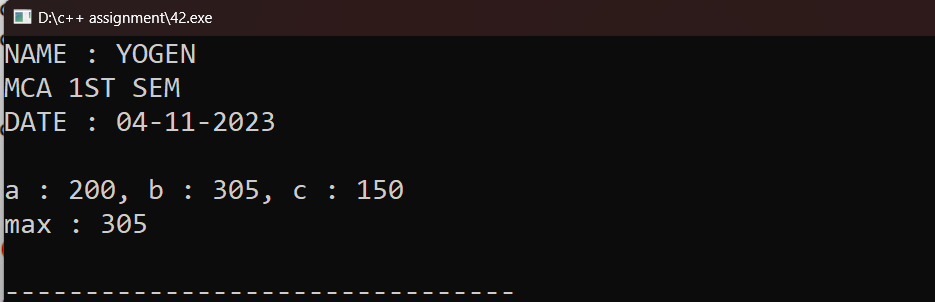
t.display() ;

t.displayMax() ;

return 0 ;

}

Output:



**Program Q.43 Write a C++ program to create class named “My\_class”, having two private member of integer type. And perform addition, multiplication, and subtraction operation inside the class body.**

**Code :**

#include<iostream>

using namespace std;

class My\_class {

int a,b; // private data members

public:

void addition(int a,int b){

cout<<"addition of "<<a<<" and "<<b<<" is = "<<a+b<<endl;

}

void subtraction(int a,int b){

cout<<"subtraction of "<<a<<" and "<<b<<" is = "<<a-b<<endl;

}

void multiplication(int a,int b){

cout<<"multiplication of "<<a<<" and "<<b<<" is = "<<a\*b<<endl;

}

void division(int a,int b){

cout<<"division of "<<a<<" and "<<b<<" is = "<<a/b<<endl;

}

};

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 04-11-2023 "<<endl<<endl;

int a,b;

cout<<"Enter two number "<<endl;

cin>>a>>b;

My\_class yogen;

yogen.addition(a,b);

yogen.subtraction(a,b);

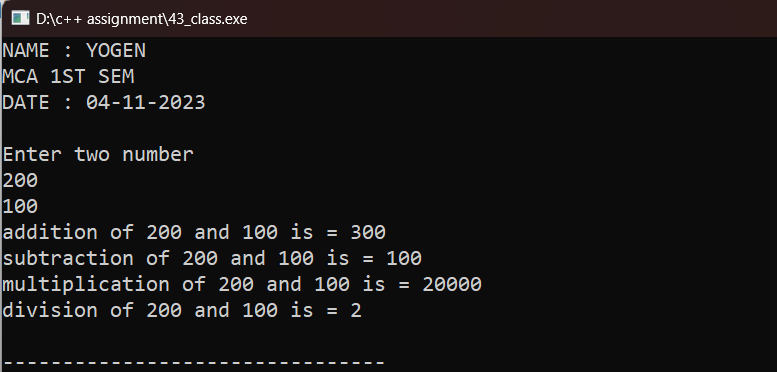
yogen.multiplication(a,b);

yogen.division(a,b);

return 0;

}

Output:

****

**Program 44 Write a C++ program to make outside function inline.**

**Code:**

#include<iostream>

using namespace std;

class InFun{

int a,b;

public:

void sum();

};

inline void InFun::sum(){ // using inline function

cout<<"Enter two number "<<endl;

cin>>a>>b;

cout<<"Sum of "<<a<<" and "<<b<<" is = "<<a+b;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 01-11-2023 "<<endl<<endl;

InFun s;

s.sum();

return 0;

}

Output:



**Program 45.Write a C++ program to keep count of created object using static member.**

**Code:**

#include<iostream>

using namespace std;

class yogen{

public:

static int a; // intialize static data member

void count(){

++a;

cout<<"No. of object is = "<<a<<endl;

}

int ObjectCount() {

return a; // Return the current count of objects

}

};

int yogen::a=0; // define static data member

int main(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 05-11-2023 "<<endl<<endl;

yogen y1;

y1.count();

yogen y2;

y2.count();

yogen y3;

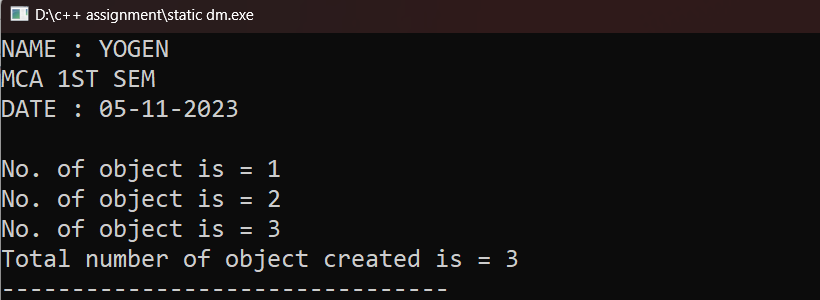
y3.count();

cout<<"Total number of object created is = "<<y1.ObjectCount();

return 0;

}

Output:



**Program 46.Define a class candidate in C++ with the following description:=**

**Private member:-**

**• A data member RNo (Registration Number) of type long.**

**• A data member Name of type string.**

**• A data member Score of type float.**

**• A data member Remarks of type string.**

**• A member function AssignRem( ) to assign Remarks as per the Score obtained by a candidate. Score range and the respective Remarks are shown as follows:**

**Score Remarks**

**>=50 Selected**

**Less than 50 Not Selected**

**Public member:-**

**• A member function Enter( ) to allow user to enter values for RNo, Name, Score and call function AssignRem( ) to assign the remarks.**

**• A member function DISPLAY( ) to allow user to view the content of all the data members.**

**Code:**

#include<iostream>

using namespace std;

class candidate{

long rno;

string name,Remarks;

float score;

void AssignRem();

public:

void Enter();

void Display();

};

void candidate::AssignRem(){

score>=50? Remarks = "SELECTED":Remarks = "NOT SELECTED";

}

void candidate::Enter(){

cout<<"Enter roll no. : ";

cin>>rno;

cout<<"Enter name : ";

getline(cin,name);

getline(cin,name);

cout<<"Enter score : ";

cin>>score;

AssignRem(); // calling private member function

}

void candidate::Display(){

cout<<"\nRoll no : "<<rno;

cout<<"\nName : "<<name;

cout<<"\nScore : "<<score

;

cout<<"\nRemarks : "<<Remarks;

}

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE : 07-11-2023 "<<endl<<endl;

candidate obj;

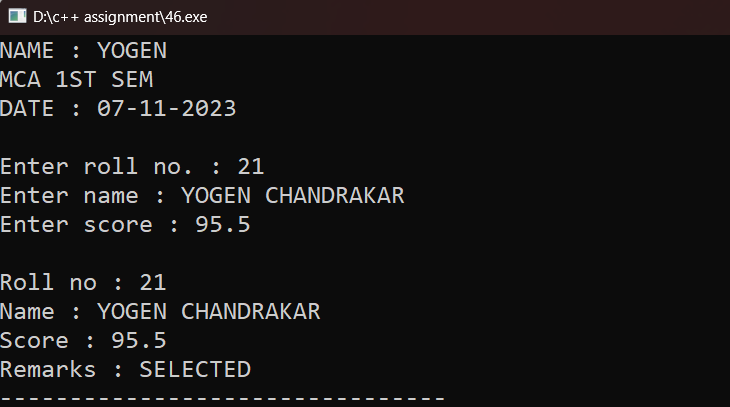
obj.Enter();

obj.Display();

return 0;

}

OUTPUT:





**Program 47.Write a C++ program to implement single inheritance.**

**Code:**

#include<iostream>

using namespace std;

class Parent{ // Base class

int rollno;

public:

void enter(){

cout<<"Enter roll no. ";

cin>>rollno;

}

void show(){

cout<<"\nRoll no.: "<<rollno<<endl;

}

};

class Child : private Parent{ // Derived class

char name[100];

public:

void input(){

enter();

cout<<"enter name ";

cin>>name;

}

void output(){

show();

cout<<"Name : "<<name<<endl;

}

};

int main()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE :08-11-2023 "<<endl<<endl;

Child c;

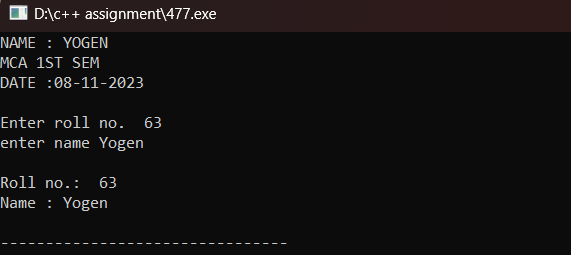
c.input();

c.output();

return 0;

}

Output:



**Program 48.Write a C++ program to implement multiple inheritance.**

**Code:**

#include<iostream>

using namespace std;

class A //base class first

{

public:

int x;

void getx()

{

cout << "enter value of x: ";

cin >> x;

}

};

class B //base class second

{

public:

int y;

void gety()

{

cout << "enter value of y: ";

cin >> y;

}

};

class C : public A, public B //C is derived from base class A and class B

{

public:

void mul()

{

cout << "multiplication is = " << x \* y;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 15-11-2023 00:05 "<<endl<<endl;

}

int main()

{

my\_info();

C obj1; //object of derived class C

obj1.getx();

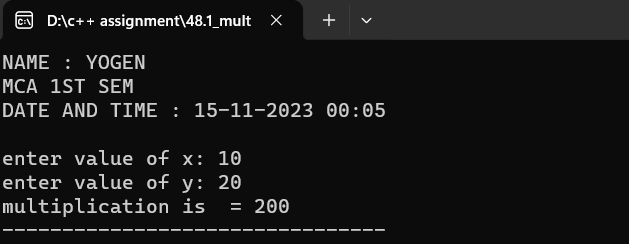
obj1.gety();

obj1.mul();

return 0;

}

Output:



**Program 49.Write a C++ program to implement multilevel inheritance.**

**Code:**

#include<iostream>

using namespace std;

class base //single base class

{

public:

int x;

void getdata()

{

cout << "Enter value of x= ";

cin >> x;

}

};

class derive1 : public base // derived class from base class

{

public:

int y;

void readdata()

{

cout << "\nEnter value of y= ";

cin >> y;

}

};

class derive2 : public derive1 // derived from class derive1

{

private:

int z;

public:

void indata()

{

cout << "\nEnter value of z= ";

cin >> z;

}

void product()

{

cout << "\nProduct= " << x \* y \* z;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 15-11-2023 00:05 "<<endl<<endl;

}

int main()

{

my\_info();

derive2 a; //object of derived class

a.getdata();

a.readdata();

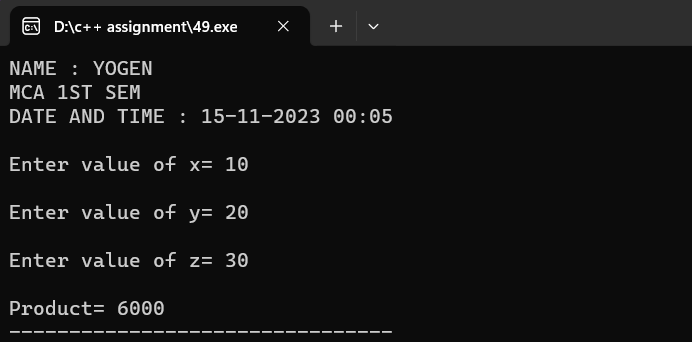
a.indata();

a.product();

return 0;

}

Output:



**Program 50. Write a C++ program to initialize three integer values through constructor using parameter.**

Code:

#include<iostream>

using namespace std;

class Number{

int a,b,c;

public:

Number(int x,int y,int z){

a = x;

b = y;

c = z;

cout<<"a is = "<<a<<"\nb is = "<<b<<"\nc is = "<<c;

}

};

void myinfo(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 23-11-23 15:25 "<<endl<<endl;

}

int main()

{

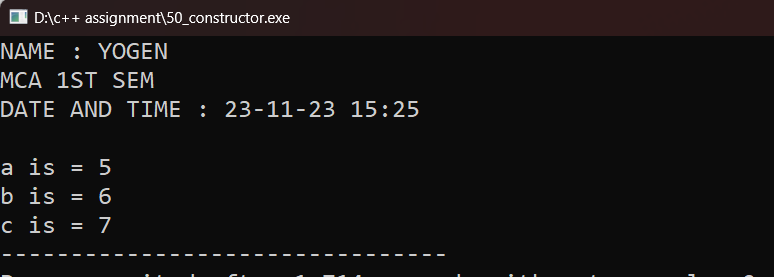
myinfo();

Number n(5,6,7);

return 0;

}

Output:



**Program 51 .Write a C++ program to illustrate order of invocation for these, Define two class first class named “Sub” and second named “Mark” in C++ with the following description:=**

**Class “Sub” contains -**

**Private member:-**

**• A data member ppr1\_code of type int.**

**• A data member ppr2\_code of type int.**

**Public member:-**

**• A default constructor which initializes and display the private member of its class.**

**Class “Mark” contains –**

**Private member:-**

**• A data member ppr1\_mark of type float.**

**• A data member ppr2\_mark of type float.**

**Public member:-**

**• A Parameterized constructor which initializes and display the private member of its class.**

create an object of Sub class inside these class.

Code:

#include<iostream>

using namespace std;

class Sub{

int ppr1\_code;

int ppr2\_code;

public:

Sub(){

ppr1\_code = 101;

ppr2\_code = 102;

cout<<"\nppr1 code : "<<ppr1\_code<<"\nppr2\_code : "<<ppr2\_code<<endl;

}

};

class Mark {

float ppr1\_mark;

float ppr2\_mark;

public:

Mark(float x , float y){ // intializing parameterized counstructer

ppr1\_mark = x;

ppr2\_mark = y;

cout<<"\nppr1 mark : "<<ppr1\_mark<<"\nppr2\_mark : "<<ppr2\_mark<<endl;

Sub s;

}

};

my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 24-11-23 11:07 "<<endl;

}

int main()

{

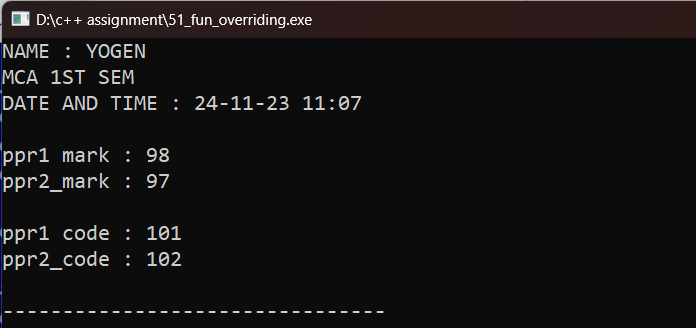
my\_info();

Mark m(98,97);

return 0;

}

Output:



**Program 52.Write a C++ program to invoke a constructor having default argument.**

**Code:**

#include<iostream>

using namespace std;

class Addition{

int a,b,c;

public :

Addition(int x=97,int y=98,int z=99) // default counstructer

{

a = x;

b = y;

c = z;

cout<<"a is "<<a<<"\nb is "<<b<<"\nC is "<<c<<endl;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 24-11-23 23:57"<<endl<<endl;

}

int main()

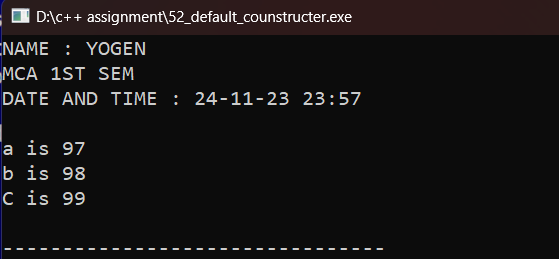
{

my\_info();

Addition a;

}

Output:



**Program 53. Write a C++ program to copy one object variable to another object using copy constructor.**

Code:

#include<iostream>

using namespace std;

class A{

int a;

public :

A(){

cout<<"enter the value "<<endl;

cin>>a;

}

A(A &y){ *// syntax : class\_name(class\_name & passing\_object)*

a = y.a;

cout<<"value of a is "<<a;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 27-11-23 21:45 "<<endl<<endl;

}

int main()

{

my\_info();

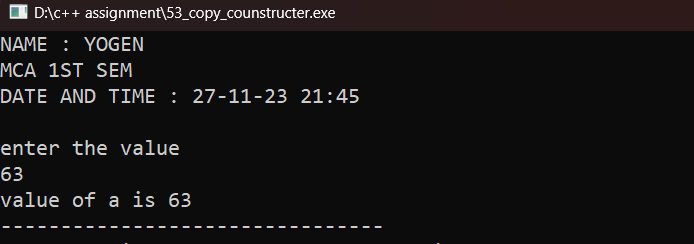
A x;

A y(x); *// calling copy counstructer*

return 0;

}

Output:



**Program 54.Write a C++ program to perform constructor overloading having three constructor within a class.**

**Code:**

#include<iostream>

using namespace std;

class cons {

int a,b,c;

public:

cons(int x=10,int y=20){

a = x;

b = y;

cout<<"1st constructor value is:\n"<<a<<endl<<b<<endl;

}

cons(int x,int y,int z){

a = x;

b = y;

c = z;

cout<<"2nd constructor value is:\n" <<a<<endl<<b << endl <<c<<endl;

}

cons(float x,float y,float z){

a = x;

b = y;

c = z;

cout<<"3rd constructor value is:\n" <<a<< endl <<b <<endl <<c<<endl;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 27-11-23 22:47 "<<endl<<endl;

}

int main({

my\_info();

cons c;

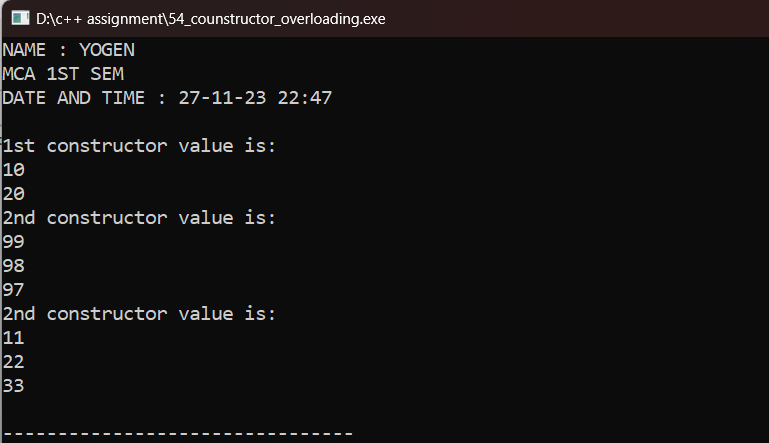
cons d(99,98,97);

cons e(11,22,33);

return 0;

}

Output:



**Program 55.Write a C++ program to allocate and deallocate memory at run time for a variable.**

**Code:**

#include<iostream>

using namespace std;

int main()

{

int \*p = new int();

\*p = 10;

cout<<"before delete value of p is "<<\*p<<"\nAnd address is "<<p<<endl;

delete p;

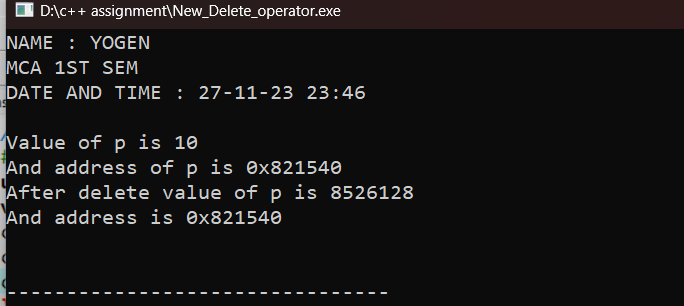
cout<<"after delete value of p is "<<\*p<<"\nafter delete address of is "<<p<<endl;

cout<<endl;

return 0;

}

Output:



**Program 56. Write a C++ program to demonstrate run time polymorphism(function overriding).**

**Code:**

#include<iostream>

using namespace std;

class Base {

public:

void print() {

cout << "This is Base Function" << endl;

}

};

class Derived : public Base {

public:

void print() { //using function overriding

cout << "This is Derived Function" << endl;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 11-12-23 16:12 "<<endl<<endl;

}

int main(){

my\_info();

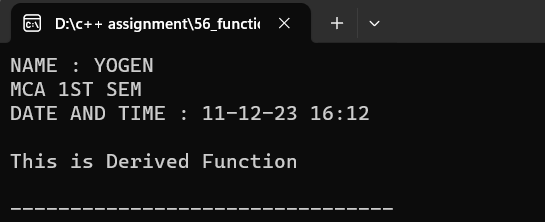
Derived derived1;

derived1.print();

return 0;

}

Output:



**Program 57. Write a C++ program to create a class named “Student” having two private member name type string and age type int. and in public section class contain one member function named “Stu\_info” which initializes the data members of its class at run time and one another member function named “Show\_info” which display the detail of a student (name,age). And invoke them using pointer to object.**

**Code:**

#include<iostream>

using namespace std;

class Student {

int age;

string name;

public:

int Stu\_info(string n,int a){

name = n;

age = a ;

}

int Show\_info(){

cout<<"The name of student is : "<<name<<endl<<"age of student : "<<age;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 09-12-23 23:52 "<<endl<<endl;

}

int main()

{

my\_info();

Student s,\*p;

p = &s;

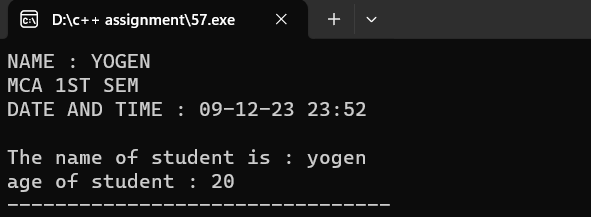
p->Stu\_info("yogen",20); //invoke using pointer

p->Show\_info();

return 0;

}

Output:



**Program 58. Write a C++ program to illustrate functioning of this pointer.**

**Code:**

#include<iostream>

using namespace std;

class A{

int a,b;

public:

void GetData(int a,int b){

this->a=a\*b;

this->b=a+b;

}

void Display(){

cout<<"multiplication is : "<<a<<endl;

cout<<"addition is : "<<b<<endl;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 10-12-23 20:25 "<<endl<<endl;

}

int main()

{

my\_info();

A a;

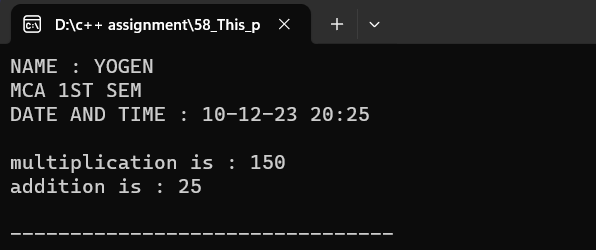
a.GetData(10,15);

a.Display();

return 0;

}

**Output:**

****

**Program 59. Write a C++ program to create two classes (named First\_class and Second\_class), and perform multiplication of two number (where first number is public data member of First\_class and second number is public data member of Second\_class) using friend function.**

**Code:**

#include<iostream>

using namespace std;

class First\_class{

public:

int a;

void get\_first(){

cout<<"Enter the first number : ";

cin>>a;

}

};

class Second\_class{

public:

int b;

void get\_second(){

cout<<"Enter the second number : ";

cin>>b;

}

friend int mul(First\_class,Second\_class); // declaring friend function

};

int mul(First\_class f,Second\_class s){ //defining friend function outside the class

int result = f.a\*s.b;

return result;

}

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 10-12-23 00:00 "<<endl<<endl;

}

int main()

{

my\_info();

First\_class l;

l.get\_first();

Second\_class m;

m.get\_second();

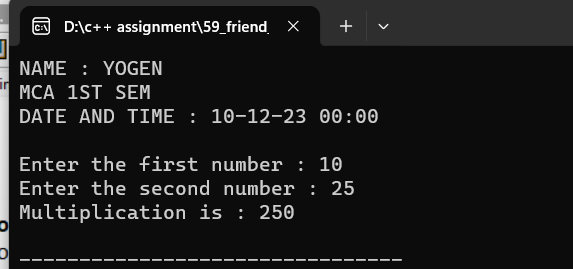
int r = mul(l,m);

cout<<"Multiplication is : "<<r<<endl;

return 0;

}

**Output:**

****

**Program 60. Write a C++ program to the working of virtual function.**

**Code:**

#include<iostream>

using namespace std;

class A{

int x=5;

public:

virtual void display(){ // create virtual function

cout<<"squre of "<<x<<" is "<<x\*x<<endl;

}

};

class B:public A{ //create derive class b

int y=10;

public:

void display(){

cout<<"square of "<<y<<" is = "<<y\*y<<endl;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : "<<endl<<endl;

}

int main(){

my\_info();

A \*a;

B b;

a = &b;

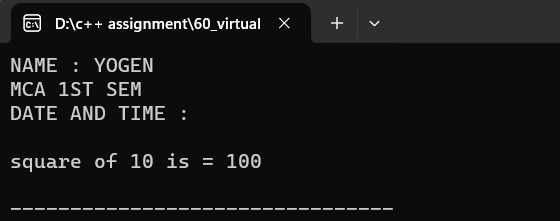
a->display();

b.display();

return 0;

}

**Output:**

****

**Program 61. Write a C++ program to the working of pure virtual function.**

**Code:**

#include<iostream>

using namespace std;

class A {

public:

int a,b;

void GetData(){

cout<<"Enter the length ";

cin>>a;

cout<<"Enter the breadth ";

cin>>b;

}

virtual void area()=0; //declaring pure virtual function

};

class B : public A{

public:

void area(){ //pure virtual function define in derived class

cout<<"Area of rectangle is "<<a\*b;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 10-12-23 12:38 "<<endl<<endl;

}

int main(){

my\_info();

A a //we can not create an object of class A because class A is a now abstract class

A \*ptr;

B b;

ptr = &b;

ptr->GetData();

ptr->area();

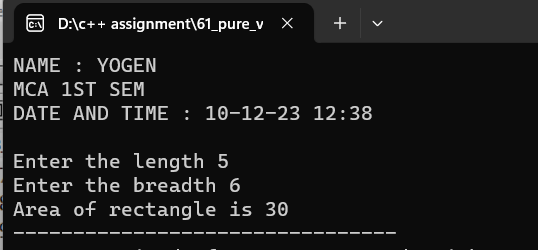
//b.GetData();

//b.area();

return 0;

}

**Output:**

****

**Program 62. Write a C++ program to find large number between two number using friend class.**

**Code:**

#include<iostream>

using namespace std;

class A{

int n1,n2;

public:

A(int i,int j)

{

n1=i;

n2=j;

}

friend class B; // declaring friend class

};

class B

{

public:

void max(A a){

if (a.n1>a.n2) {

cout<<a.n1<<"greater number ";

}

else {

cout<<a.n2<<" is greater number ";

}

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : "<<endl<<endl;

}

int main(){

my\_info();

A a(100,200);

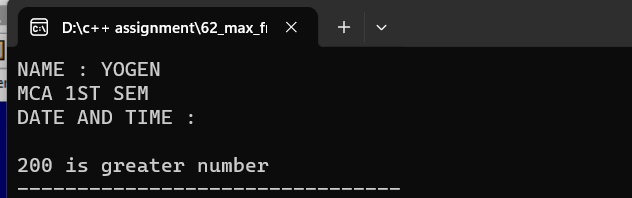
B b;

b.max(a);

return 0;

}

**Output:**

****

**Program 63. Write a C++ program to for operator overloading to compare two objects are equal are not using = =operator.**

**Code:**

**// code :**

#include<iostream>

using namespace std;

class First\_class{

int number;

public:

First\_class(int num){

number = num;

cout<<"Number is : "<<number<<endl;

}

First\_class operator ==(First\_class obj)

{

if(number==obj.number)

{

cout<<"Both are equal "<<endl;

}

else

{

cout<<"Both are not equal "<<endl;

}

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 10-12-2023 04:05 "<<endl<<endl;

}

int main()

{

my\_info();

First\_class F1(5),F2(3) ,F3(3);

cout<<"Comaparison between the first and second object : "<<endl;

F1==F2;

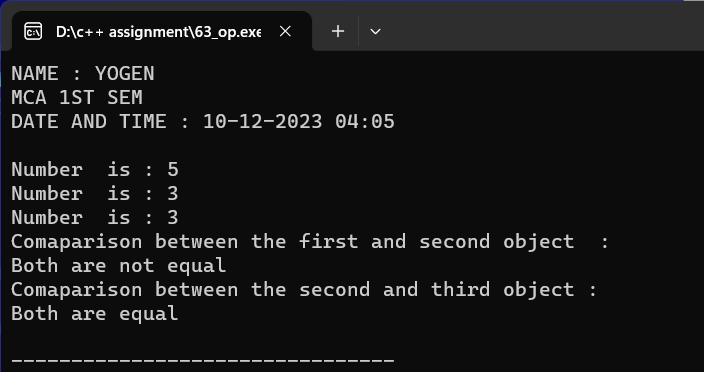
cout<<"Comaparison between the second and third object : "<<endl;

F2==F3;

return 0;

}

**Output:**

****

**Program 64. Write a C++ program to illustrate unary operator overloading on increment operator by using member function.**

**Code:**

//member function.

#include<iostream>

using namespace std;

class yogen{

int a,b;

public:

yogen(int x){

a=x;

cout<<"Before increment value of a is "<<a<<endl;

}

void operator ++(){

a=++a;

cout<<"After increment value of a is "<<a<<endl;

}

};

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 10-12-2023 00:05 "<<endl<<endl;

}

int main()

{

my\_info();

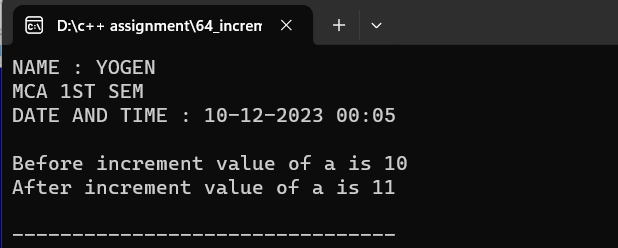
yogen y(10);

++y;

return 0;

}

**Output:**

****

**Program 65. Write a C++ program to add two complex number using + operator overloading by friend function.**

**Code:**

#include<iostream>

using namespace std;

class add\_two\_complex {10

655

int a,b,c,d;

public:

void firstcomplex(){

cout<<"Enter the value of a and b : "<<endl;

cin >>a >> b;

cout<<"First complex number is : "<<endl;

cout<<a<<"+"<<b<<"i"<<endl;

}

void secondcomplex(){

cout<<"Enter the value of c and d : "<<endl;

cin>>c>>d;

cout<<"Second complex number is : "<<endl;

cout<<c<<"+"<<d<<"i"<<endl;

}

friend void operator +(add\_two\_complex); //declare friend function

};

void operator + (add\_two\_complex s1)

{

int n,m;

n = s1.a+s1.c;

m = s1.b+s1.d;

cout<<"addition is = "<<n<<" + "<<m<<"i"<<endl;

}

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 11-12-2023 20:07 "<<endl<<endl;

}

int main()

{

my\_info();

add\_two\_complex s1;

s1.firstcomplex();

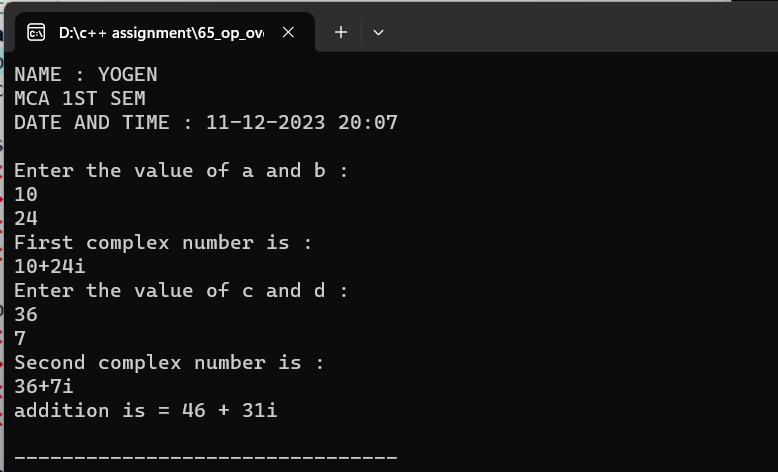
s1.secondcomplex();

+s1;

return 0;

}

**Output:**

****

**Program 66. Write a C++ program to illustrate unary minus operator overloading using friend function.**

**Code:**

#include<iostream>

using namespace std;

class overload

{

int a;

public:

void get()

{

cout<<"Enter the number : ";

cin>>a;

cout<<"Before overloading value of a is := "<<a<<endl;

}

friend void operator -(overload ); // Declaring friend function with operator overloading

};

void operator -(overload x)

{

x.a=-x.a;

cout<<"After overloading value of a is := "<<x.a<<endl;

}

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 11-12-2023 20:15 "<<endl;

cout<<"\*-------\*--------\*"<<endl;

}

int main()

{

my\_info();

overload o;

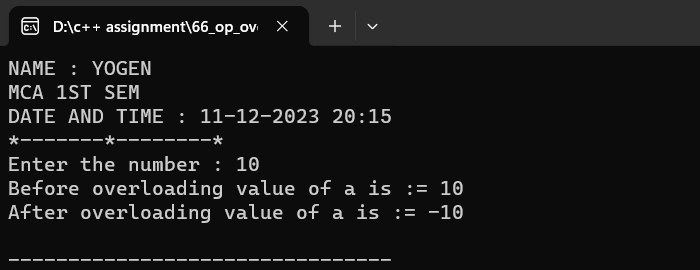
o.get();

-o; // unary minus operator

return 0;

}

**Output:**

****

**Program 67. WAP to demonstrate the use of >> and getline( ) for reading the string.**

**Code:**

#include<iostream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 13-12-2023 11:30 "<<endl<<endl;

}

void str(){

string Fname,Lname ;

cout<<"Enter your first name : ";

cin>>Fname;

cout<<"Enter your last name : ";

getline(cin,Lname);

getline(cin,Lname);

cout<<"The name of student is : "<<Fname<<" "<<Lname;

}

int main()

{

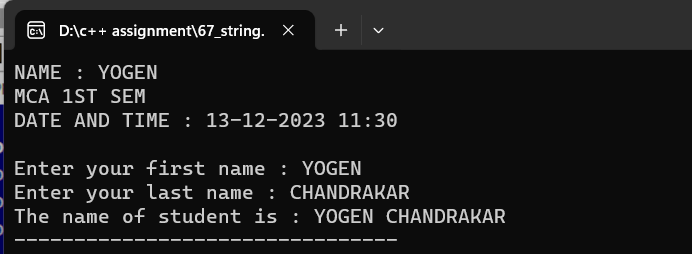
my\_info();

str();

return 0;

}

**Output:**

****

**Program 68. WAP to create a file named “Rudra” using constructor.**

**Code:**

#include <iostream>

#include <fstream>

using namespace std;

class CreateFile {

public:

CreateFile() { // Constructor to create a file named "Rudra"

ofstream file("Rudra.txt");

if (file.is\_open())

{

cout<<"File named 'Rudra.txt' created successfully!" << std::endl;

// Closing the file

file.close();

}

else

{

cout<<cerr << "Error creating the file!" << std::endl;

}

}

};

void my\_info()

{

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 13-12-2023 11:38 "<<endl<<endl;

}

int main(){

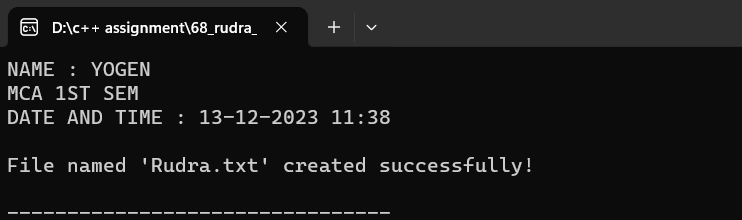
my\_info();

CreateFile c;

return 0;

}

**Output:**

****

**Program 69. WAP to create a file name “Info” using open function having details about your name,age,class and address.and display them into the console using eof( ).**

**Code:**

#include <iostream>

#include <fstream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 13-12-2023 00:01 "<<endl<<endl;

}

int main() {

my\_info();

ofstream outFile("Info.txt");

if (outFile.is\_open()) {

string name, address;

int age;

char className[50];

cout << "Enter your name: ";

getline(cin, name);

cout << "Enter your age: ";

cin >> age;

cin.ignore();

cout << "Enter your class: ";

cin.getline(className, 50);

cout << "Enter your address: ";

getline(cin, address);

outFile << "Name: " << name << endl;

outFile << "Age: " << age << endl;

outFile << "Class: " << className << endl;

outFile << "Address: " << address << endl;

outFile.close();

cout << "File 'Info.txt' created successfully!" << endl << endl;

ifstream inFile("Info.txt");

if (inFile.is\_open()) {

cout << "Reading contents from 'Info.txt':" << endl;

while (!inFile.eof()){

string line;

getline(inFile, line);

cout << line << endl;

}

inFile.close();

}

else

{

cout << "Error opening the file for reading!" << endl;

}

}

else

{

cout << "Error creating the file!" << endl;

}

return 0;

}

**Output:**

****

**Program 70.WAP to perform truncate operation in a file existing file named “File”.**

**Code:**

#include <iostream>

#include <fstream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 00:05 "<<endl<<endl;

}

void truncate(){

ofstream outFile("File", ios::trunc);

if (outFile.is\_open())

{

cout << "file 'File' has been truncated (content cleared)." << endl;

outFile.close();

}

else

{

cout << "Error opening the file for truncation!" << endl;

}

}

int main() {

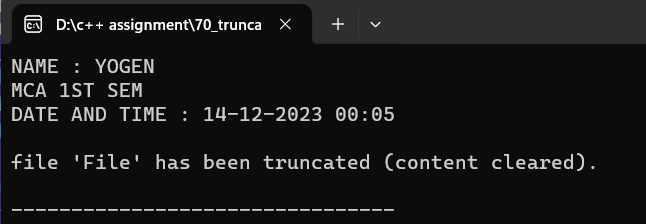
my\_info();

truncate();

return 0;

}

**Output:**

****

**Program 71.WAP to open a existing file name “Shiva” in append mode to add some content in a file.**

**Code:**

#include <iostream>

#include <fstream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 00:25 "<<endl<<endl;

}

int main() {

my\_info();

ofstream outFile("Shiva",ios::app);

if (outFile.is\_open()) {

cout << "File 'Shiva' opened in append mode." << endl;

string additionalContent;

cout << "Enter additional content to append to the file: ";

getline(cin,additionalContent);

outFile << additionalContent << endl;

cout <<"Content added to the file." << endl;

outFile.close();

}

else {

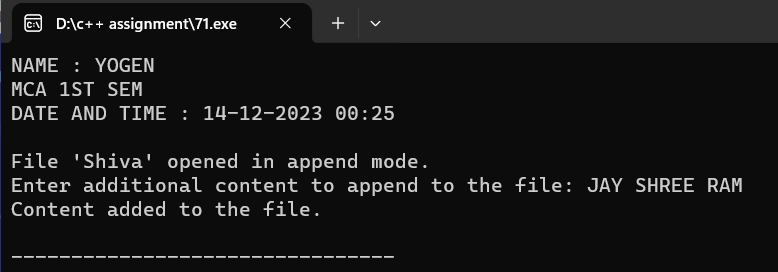
cout << "Error opening the file for appending!" << endl;

}

return 0;

}

**Output:**

****

**Program 72.WAP to find current position of input/output pointer of a file.**

**Code:**

#include <iostream>

#include <fstream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 00:05 "<<endl<<endl;

}

void curr\_pos()

{

ifstream fin("File");

cout<<"Current position of input pointer: " << fin.tellg() << endl;

ofstream fout;

fout.open("File",ios::app);

fout<<endl<<"aaj kare so ab";

cout << "Current position of output pointer: " << fout.tellp() << endl;

fin.close();

fout.close();

}

int main()

{

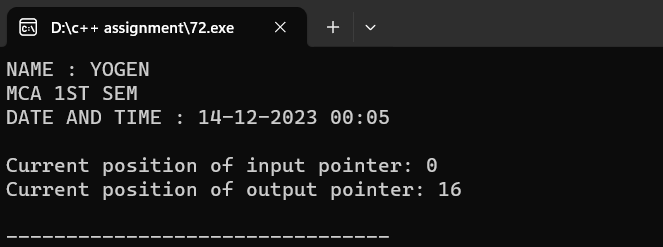
my\_info();

curr\_pos();

return 0;

}

**Output:**

****

**Program 73.WAP to differentiate read( ) and getline( ) function.**

**Code:**

//" getline()

#include<iostream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 00:40 "<<endl<<endl;

}

int main()

{

my\_info();

char name[20];

cout<<"Enter the name : ";

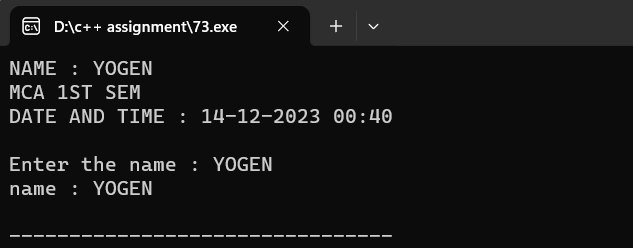
cin.getline(name,18); // read maximum 18 character

cout<<"name : "<<name<<endl;

return 0;

}

**Output:**

****

//" read()

#include<iostream>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 00:45 "<<endl<<endl;

}

int main()

{

my\_info();

char name[20];

cout<<"Enter the name : ";

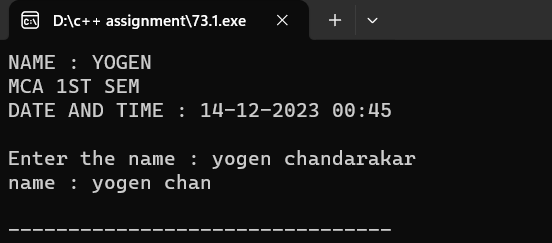
cin.read(name,10);

cout<<"name : "<<name<<endl;

return 0;

}

Output:

****

**Program 74.WAP to demonstrate manupulators(setw, setprecision, setbase, setfill).**

**Code:**

#include <iostream>

#include <iomanip>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 01:05 "<<endl<<endl;

}

int main()

{

cout<<setw(10)<<"Name: "<<setw(20)<< "YOGEN CHANDRAKAR " << endl;

cout<<setw(10)<<"Age: "<<setw(20)<<20<<endl<<endl;

double pi = 3.141592653589793;

cout<<fixed<<setprecision(4)<<"Pi (4 decimal places): "<<pi<<endl<<endl;

int number = 273;

cout<<setbase(16)<<"Hexadecimal representation of 273: "<<number<<endl<<endl;

cout<<"Original width: "<<setw(10)<<"Hello"<<endl;

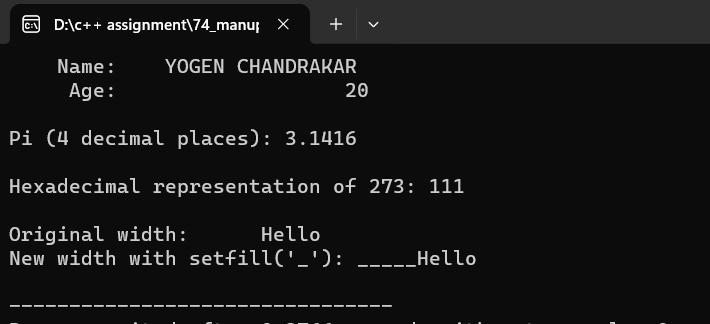
cout << setfill('\_') << "New width with setfill('\_'): " << setw(10) << "Hello" << endl;

cout << setfill(' '); // Reset fill character to space

return 0;

}

**Output:**

****

**Program 75.WAP which reads input from the keyboard whose width specified with 8 and unused space filled with '#'and input should be left-justified**

**Code:**

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

void my\_info(){

cout<<"NAME : YOGEN "<<endl;

cout<<"MCA 1ST SEM "<<endl;

cout<<"DATE AND TIME : 14-12-2023 00:40 "<<endl<<endl;

}

int main(){

my\_info();

cout << "Enter a string: ";

string input;

getline(cin, input);

if (input.length() < 8) {

input.insert(0, 8 - input.length(), '#');

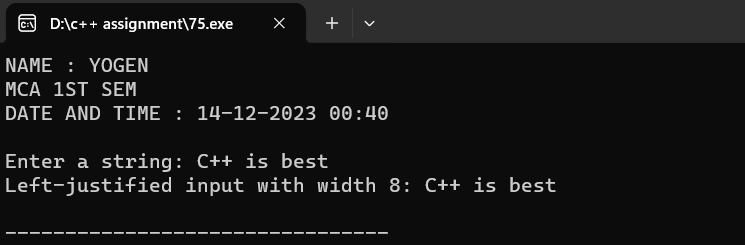
}

cout << "Left-justified input with width 8: " << input << endl;

return 0;

}

**Output:**

****