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

Code :
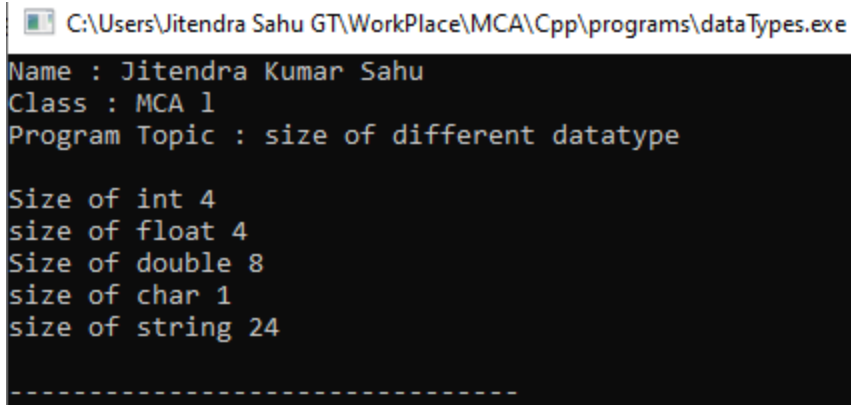
```cpp
#include <iostream>
using namespace std;


void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}
int main(){
        printIntro("Square of integer","28-09-23 13:57") ;


        int x ;
        cout << "Enter an integer : ";
        cin>> x ;
        cout << x << " cube = " << x*x*x << endl ;
        return 0 ;
}
```

Output:

Name : Jitendra Kumar Sahu
Class : MCA 1
Program Topic : size of different datatype

Size of int 4
size of float 4
Size of double 8
size of char 1
size of string 24

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntroWithTopic(string programTopic){
        cout <<"Name : Jitendra Kumar Sahu"<< endl ;
        cout << "Class : MCA l" << endl ;
        cout <<"Program Topic : " << programTopic << endl << endl ;
}
int main(){
        printIntroWithTopic("Arithemetic Operators ")  ;
        int a ,b ;
        cout << "Enter value of a and b : "  ;
        cin >> a >> b ;
        cout << "addition Operator, a + b : " << a+b << endl ;
        cout << "substraction Operator, a - b :  " << a-b << endl ;
        cout << "multiplication Operator, a * b : " << a*b << endl ;
        cout << "divid Operator, a / b : " << a/b << endl ;
        cout << "modulo Operator, a % b : " << a%b << endl ;
        cout << "pre increament Operator, ++a : " << ++a << endl ;
        cout << "post increament Operator, a++ : " << a++ << endl ;
        cout << "pre decreament Operator, --b : " << --b << endl ;
        cout << "post decreament Operator, b-- : " << b-- << endl ;
        return 0 ;
}
```

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\arithmetic-operators.exe

Name : Jitendra Kumar Sahu
Class : MCA l
Program Topic : Arithemetic Operators

Enter value of a and b : 4 2
addition Operator, a + b : 6
substraction Operator, a - b :  2
multiplication Operator, a * b : 8
divid Operator, a / b : 2
modulo Operator, a % b : 0
pre increament Operator, ++a : 5
post increament Operator, a++ : 5
pre decreament Operator, --b : 1
post decreament Operator, b-- : 1
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntroWithTopic(string programTopic){
        cout <<"Name : Jitendra Kumar Sahu"<< endl ;
        cout << "Class : MCA l" << endl ;
        cout <<"Program Topic : " << programTopic << endl << endl ;
}
int main(){
        printIntroWithTopic("Relational Operators") ;
        int a, b  ;
        cout << "0 = FALSE\n1 = TRUE\n"  ;
        cout << "Enter values for a and b " ;
        cin >> a >> b ;
        cout <<a << " == "<<b <<" : "<< (a==b) << endl ;
        cout <<a << " != "<<b <<" : "<< (a!=b) << endl ;
        cout <<a << " > "<<b <<" : "<< (a>b) << endl ;
        cout <<a << " < "<<b <<" : "<< (a<b) << endl ;
        cout <<a << " >= "<<b <<" : "<< (a>=b) << endl ;
        cout <<a << " <= "<<b <<" : "<< (a<=b) << endl ;
        return 0 ;
}
```



```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\relationalOperator.exe

Name : Jitendra Kumar Sahu
Class : MCA l
Program Topic : Relational Operators

0 = FALSE
1 = TRUE
Enter values for a and b 4 5
4 == 5 : 0
4 != 5 : 1
4 > 5 : 0
4 < 5 : 1
4 >= 5 : 0
4 <= 5 : 1

--------------------------------
```

Program04: Write a C++ program to demonstrate the logical operators(&&,||,!) .
Code :

```cpp
#include <iostream>
using namespace std ;

void printIntroWithTopic(string programTopic){
        cout <<"Name : Jitendra Kumar Sahu"<< endl ;
        cout << "Class : MCA l" << endl ;
        cout <<"Program Topic : " << programTopic << endl << endl ;
}

int main(){
        printIntroWithTopic("Logical Operators") ;

        bool  a  = true , b = false ;

        cout << a << " && "<< b <<" : " << (a && b) << endl ;
        cout << a << " || "<< b <<" : "<< (a || b) << endl ;
        cout << "!"<< a  << " : "<< !b << endl ;

        return 0 ;
}
```
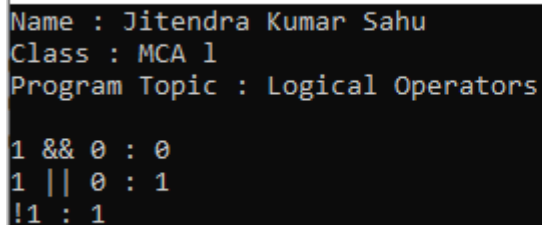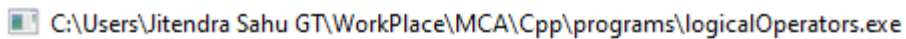
```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\logicalOperators.exe

Name : Jitendra Kumar Sahu
Class : MCA l
Program Topic : Logical Operators

1 && 0 : 0
1 || 0 : 1
!1 : 1

--------------------------------
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(){
      cout <<"Name: Porogram to demonstrate Bitwise Operator "<< endl ;
      cout << "Author: Jitendra Kumar Sahu" << endl ;
      cout <<"Date: 28-09-23 13:03" << endl << endl;
}
void bitwise(int a , int b){
      int res = a & b ;
      cout << "a & b : " << res << endl ;

      res = a | b ;
      cout << "a | b : " << res << endl ;

      res = a << b ;
      cout << "a << b : " << res << endl ;

      res = a >> b ;
      cout << "a >> b : " << res << endl ;

      res = ~b ;
      cout << "~b : " << res << endl ;

}

int main(){
      printIntro() ;

      int a, b ;
      cout << "Enter the value of a and b : " ;
      cin >> a >> b ;
      bitwise(a,b) ;
      return 0 ;
}
```
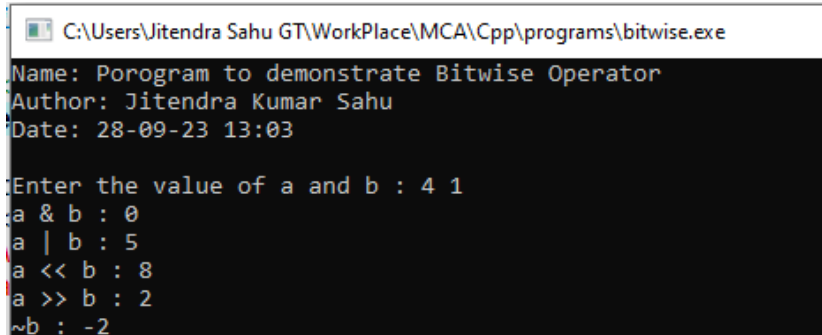
Output:

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\bitwise.exe

Name: Porogram to demonstrate Bitwise Operator
Author: Jitendra Kumar Sahu
Date: 28-09-23 13:03

Enter the value of a and b : 4 1
a & b : 0
a | b : 5
a << b : 8
a >> b : 2
~b : -2
```

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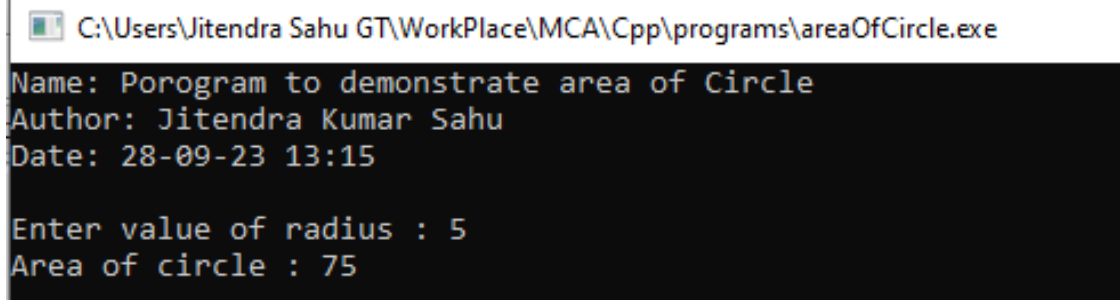
Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(){
        cout <<"Name: Porogram to demonstrate area of Circle "<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Date: 28-09-23 13:15" << endl << endl;
}
void area(int radius){
         const float PI = 22/7 ;
         cout <<"Area of circle : " << PI*radius*radius;
}
int main(){
        printIntro() ;
        int radius;
        cout << "Enter value of radius : " ;
        cin >>radius ;
        area(radius) ;
        return 0 ;
}
```

Output:


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\areaOfCircle.exe

Name: Porogram to demonstrate area of Circle
Author: Jitendra Kumar Sahu
Date: 28-09-23 13:15

Enter value of radius : 5
Area of circle : 75
```
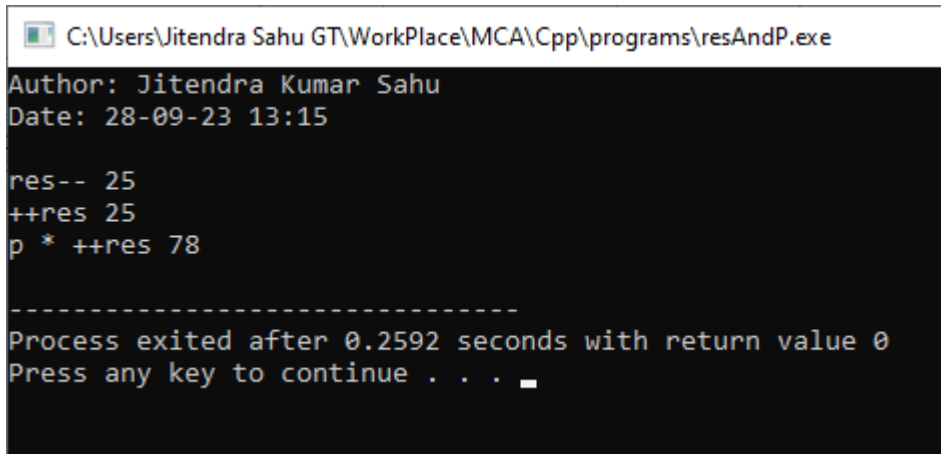
Program 07: Write a C++ program for Assuming that res starts with the value 25 and p with 3,so print the following code:-

> **cout<<res--;**
> **cout<<++res;**
> **p=p*++res;**

code :

```
#include<iostream>
using namespace std;

void printIntro(){
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Date: 28-09-23 13:15" << endl << endl;
}

int main()
{
        printIntro() ;
        int res = 25 , p = 3 ;
        cout << "res-- "<<res--<< endl ;
        cout << "++res "<<++res << endl  ;
        p = p * ++res ;
        cout << "p * ++res " << p << endl ;
        return 0;
}
```

Output :

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(){
        cout <<"Name: Week day with switch case "<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Date: 28-09-23 13:15" << endl << endl;
}
int main()
{
        printIntro() ;
        int n  ;
        cout << "Enter day of the week " ;
        cin>> n ;
        switch (n)
        {
                case 1 : cout << "Monday" << endl ;
                break ;
                case 2 : cout <<"Tuesday"<< endl ;
                break ;
                case 3 : cout <<"Wednessday" << endl ;
                break ;
                case 4 : cout <<"Thursday"<< endl ;
                break ;
                case 5 : cout <<"Friday"<< endl ;
                break ;
                case 6 : cout <<"Saturday"<< endl ;
                break ;
                case 7 : cout <<"Sunday"<< endl ;
                break ;
                default : cout << "You may have entered a wrong day" ;
        }
        return 0 ;
}
```

Output:

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\dayOfWeek.exe

```
Name: Week day with switch case
Author: Jitendra Kumar Sahu
Date: 28-09-23 13:15

Enter day of the week 2
Tuesday

----------------------------
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(){
        cout<<"Topic : basic arithmetic with switch case "<<endl;
        cout<<"Author: Jitendra Kumar Sahu"<<endl;
        cout<<"MCA 1st sem"<<endl;
        cout<<"date: 01-10-23 23:36"<<endl;
}
int main()
{
        printIntro() ;
        float n1, n2 , result;
        char operation ;

        cout << "Enter calculation " ;
        cin >> n1 >> operation >> n2 ;
        bool isValidOperation = true ;

        switch (operation) {
                case '+' : result = n1 + n2 ;
                break ;

                case '-' : result = n1 - n2 ;
                break ;

                case '/' : result = n1 / n2 ;
                break ;

                case '*' : result = n1 * n2 ;
                break ;

                default : cout << "You may have entered an invalid operation"<< endl;
                isValidOperation = false ;
        }

        if (isValidOperation) cout << "result " << result << endl ;
}
```
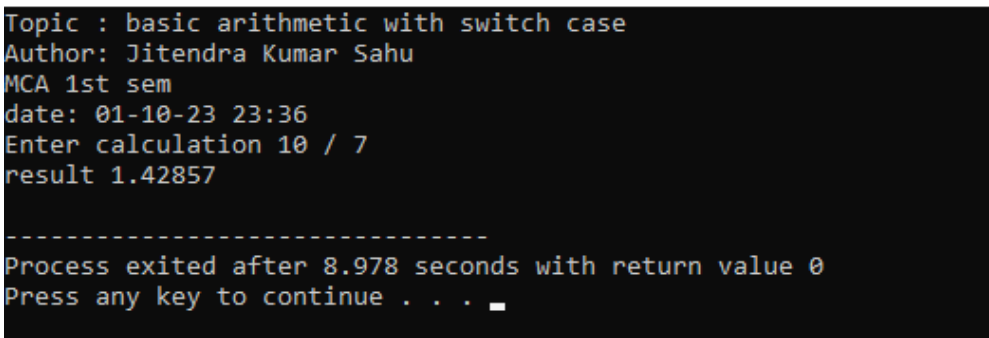
Output:

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\basicArithmetic.exe

Topic : basic arithmetic with switch case
Author: Jitendra Kumar Sahu
MCA 1st sem
date: 01-10-23 23:36
Enter calculation 10 / 7
result 1.42857


------------------------------
Process exited after 8.978 seconds with return value 0
Press any key to continue . . .
```
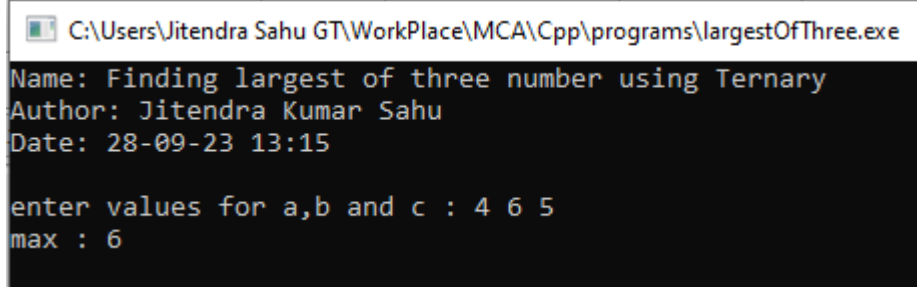
Program 10: Write a C++ program to find largest number among three numbers using ternary Operator.

Code :

```cpp
#include <iostream>
using namespace std ;
void printIntro(){
        cout <<"Name: Finding largest of three number using Ternary"<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Date: 28-09-23 13:15" << endl << endl;
}

int main()
{
        printIntro() ;

        int a, b ,c  ;
        cout << "enter values for a,b and c : " ;
        cin>> a>> b >> c ;
        int max = (a>b && a>c)?a:((b>c)?b:c );
        cout << "max : " << max << endl ;

        return 0 ;
}
```

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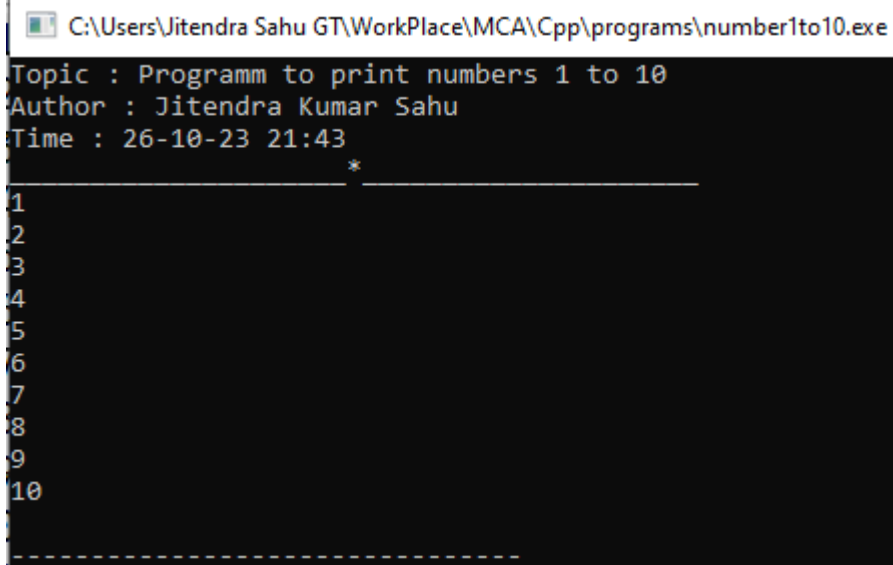
Program11: Write a C++ program using for loop to print numbers from 1 to 10.
Code :

```cpp
#include <iostream>
using namespace std ;


void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}
int main(){
        printIntro("Programm to print numbers 1 to 10","26-10-23 21:43") ;
                for(int i =1  ; i<=10 ; i++ )
                cout << i << endl ;
        return 0 ;
}
```
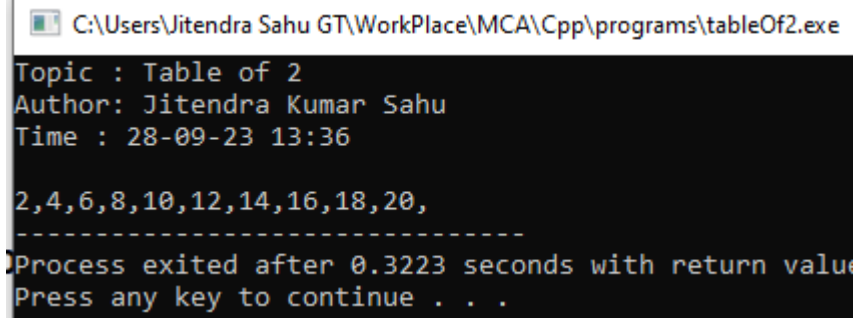
```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\number1to10.exe
Topic : Programm to print numbers 1 to 10
Author : Jitendra Kumar Sahu
Time : 26-10-23 21:43
                            *
_____ _____
1
2
3
4
5
6
7
8
9
10

--------------------------------
```

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

```cpp
#include <iostream>
using namespace std ;
void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}

void printTable(int n){
        int i = 1 ;
        while(i <= 10 ){
                        cout << i *  n <<"," ;
                        i++ ;
        }
}
int main(){
        printIntro("Table of 2","28-09-23 13:36") ;
        printTable(2) ;
        return 0 ;
}
```

Output :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\tableOf2.exe

Topic : Table of 2
Author: Jitendra Kumar Sahu
Time : 28-09-23 13:36

2,4,6,8,10,12,14,16,18,20,
------------------------------
Process exited after 0.3223 seconds with return value
Press any key to continue . . .
```

Program 13 : Write a C++ program to print following patterns-

* * * *	b)	*	c)	1	d)	*
* * *		* *		1 2		* *
* *		* * *		1 2 3		* * *
*		* * * *		1 2 3 4		* * * *

Code :

```cpp
#include <iostream>
using namespace std;

void triangle1(int n){
   for (int i = 0; i < n; i++){
      for (int j = 0; j <= i; j++){
         cout << "* ";
      }
      cout << endl;
   }
}

void triangle2(int n){
   for (int i = n; i > 0; i--){
     for (int j=0 ; j < i ; j++)
      cout << "* " ;
      cout << endl;
   }
}

void triangle3(int n){
   for(int i = 1 ; i <= n ; i++){
      for (int j = 0; j <= n; j++){
         j <= n - i ? cout <<"  " : cout <<"* " ;
      }
      cout << endl ;
   }
}

void numericPattern(int n){
   for (int i = 1 ; i<= n ; i++ ){
     for(int j = 1 ; j <= i ; j++ )
        cout << j << " " ;
        cout << endl ;
   }
}
```

```cpp
 int main()
 {
     int n;
         cout << "Enter the n : ";
         cin >> n;
         cout << "pattern A \n" ;
         triangle2(n) ;
         cout << endl ;
         cout << "pattern B \n" ;
         triangle1(n) ;
        cout << endl ;

        cout << "pattern C \n" ;
        numericPattern(n) ;
        cout << endl ;

        cout << "pattern D \n" ;
        triangle3(n) ;
   cout << endl ;
   return 0;
 }
```

Output :

```
C:\Windows\System32\cmd.exe

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>g++ sta

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>a.exe
Enter the n : 4
pattern A
* * * *
* * *
* *
*

pattern B
*
* *
* * *
* * * *

pattern C
1
1 2
1 2 3
1 2 3 4

pattern D
      *
     * *
    * * *
   * * * *
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}

int main(){
        printIntro("Square of integer","28-09-23 13:57") ;

        int x ;
        cout << "Enter an integer : ";
        cin >> x ;


        for(int i = 1 ; i <= x ; i++)
                cout << i << " cubed " << i*i*i << endl ;

        return 0 ;
}
```
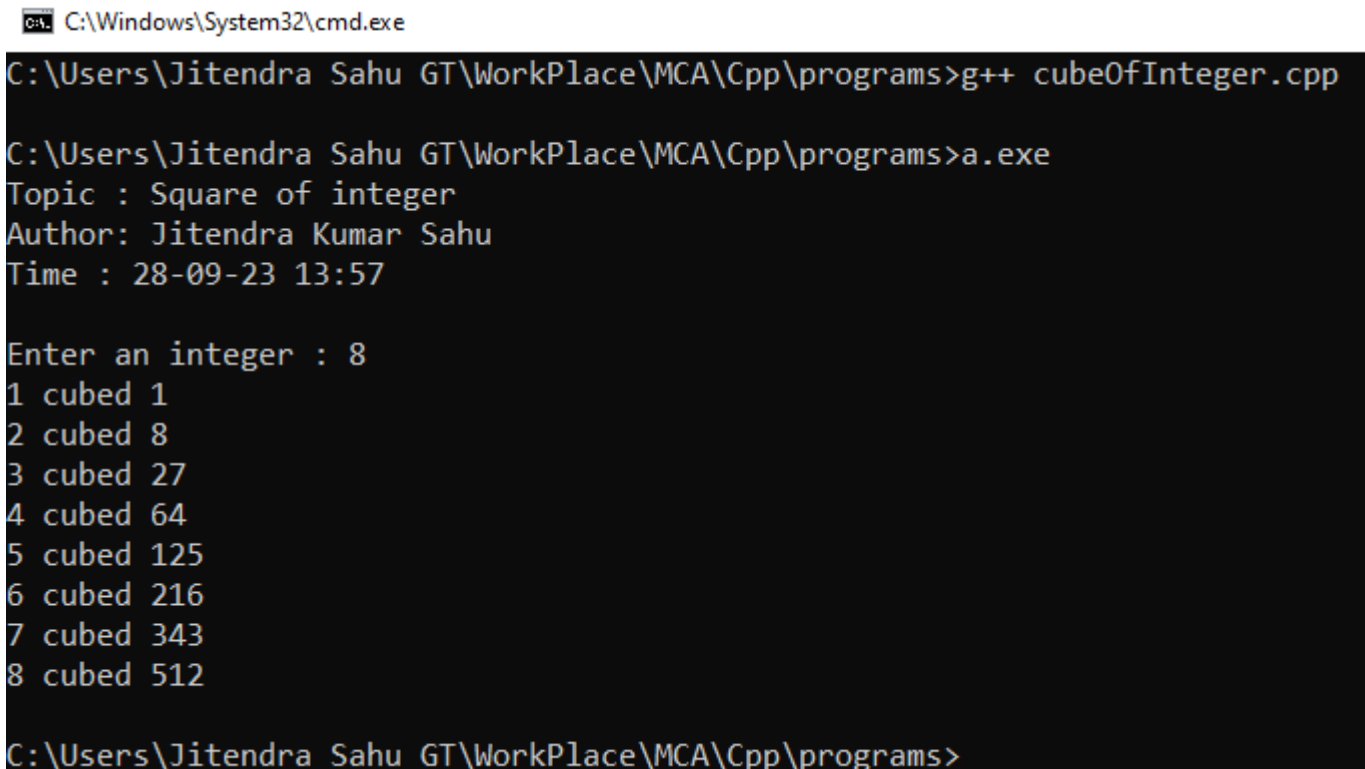
Ouput :

```
C:\Windows\System32\cmd.exe

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>g++ cubeOfInteger.cpp

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>a.exe
Topic : Square of integer
Author: Jitendra Kumar Sahu
Time : 28-09-23 13:57

Enter an integer : 8
1 cubed 1
2 cubed 8
3 cubed 27
4 cubed 64
5 cubed 125
6 cubed 216
7 cubed 343
8 cubed 512


C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>
```

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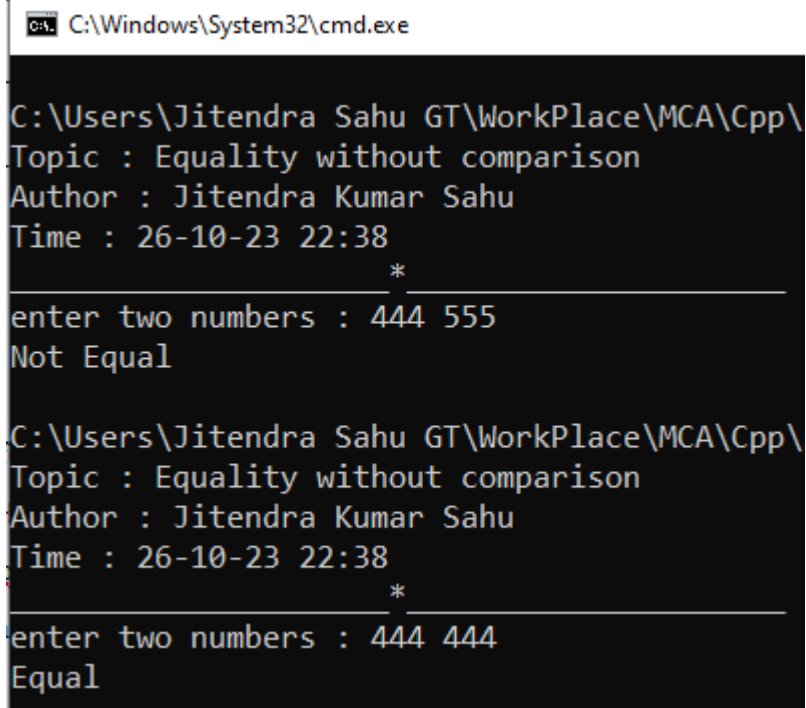
Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}

int main(){

        printIntro("Equality without comparison","26-10-23 22:38") ;

        int a,b  ;
        cout << "enter two numbers : " ;
        cin >> a >>b ;
        a ^ b ? cout << "Not Equal"    : cout<< "Equal"  ;
        cout << endl ;
        return 0 ;
}
```
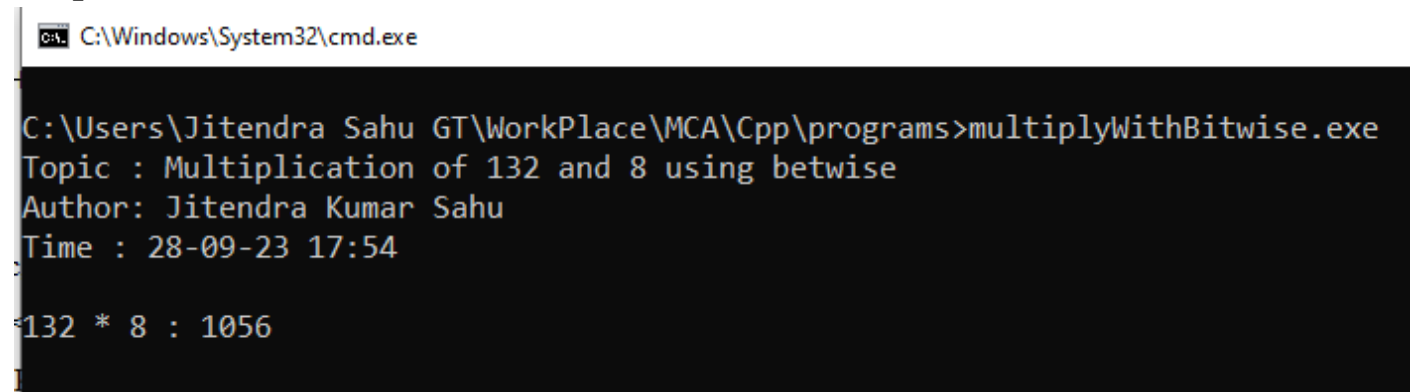
Output:

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

Code :

```cpp
#include <iostream>
using namespace std ;
void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}

int main(){
        printIntro("Multiplication of 132 and 8 using bitwise", "28-09-23 17:54") ;
        cout << (132 << 3) << endl  ;
        return 0 ;
}
```

Output:


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>multiplyWithBitwise.exe
Topic : Multiplication of 132 and 8 using betwise
Author: Jitendra Kumar Sahu
Time : 28-09-23 17:54

132 * 8 : 1056
```
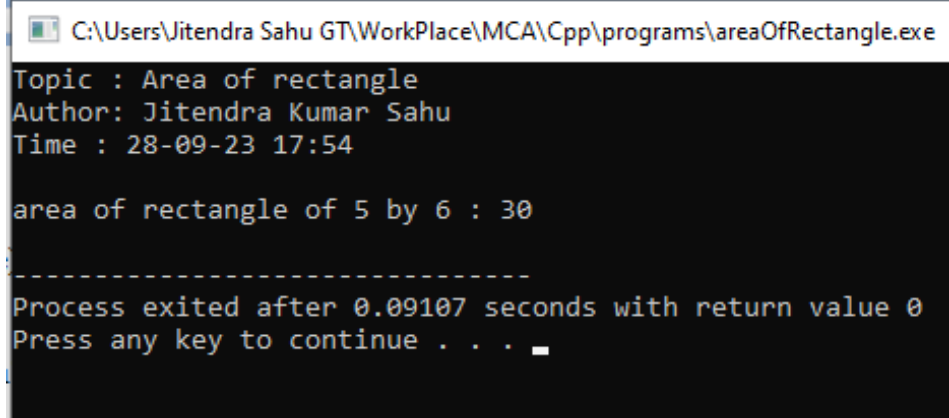
Program 17: Write a C++ program to find Area of rectangle (using #define).

Code :

```cpp
#include <iostream>
using namespace std ;

#define length 5
#define width 6
#define area (length * width )

void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}

int main(){
        printIntro("Area of rectangle", "28-09-23 17:54") ;
        cout << "area of rectangle of 5 by 6 : "  ;
        cout << area << endl ;
        return 0 ;
}
```

Output


```
Topic : Area of rectangle
Author: Jitendra Kumar Sahu
Time : 28-09-23 17:54

area of rectangle of 5 by 6 : 30

------------------------------
Process exited after 0.09107 seconds with return value 0
Press any key to continue . . .
```
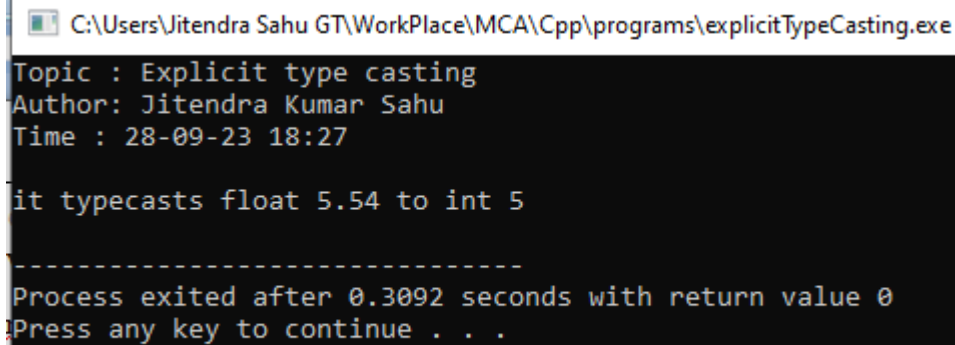
Program 18: Write a C++ program to demonstrate explicitly typecasting..

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}

int main(){
        printIntro("Explicit type casting", "28-09-23 18:27");
        float x = 5.54 ;
        int casted = (int)x ;
        cout << "it typecasts float "<< x << " to int " << casted << endl ;
        return 0 ;
}
```

Output:

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\explicitTypeCasting.exe

Topic : Explicit type casting
Author: Jitendra Kumar Sahu
Time : 28-09-23 18:27

it typecasts float 5.54 to int 5


--------------------------------
Process exited after 0.3092 seconds with return value 0
Press any key to continue . . .
```

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 numbers.

Code :

```cpp
#include<iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____\n"<< endl ;
}

int main()
{
        printIntro("sum even and odds till 20","27-10-13 08:51") ;

        int numberOfDigits = 10 ;
        int OddStart = 1 , EvenStart = 2, OddResult = 0 ,  EvenResult =0;

        while (numberOfDigits--){

                OddResult += OddStart ;
                EvenResult += EvenStart ;

                OddStart +=2 ;
                EvenStart += 2;
        }
        cout << "Sum of odd numbers till 20 : "<< OddResult<<endl  ;
        cout << "Sum of even numbers till 20 : "<< EvenResult<<endl  ;
}
```
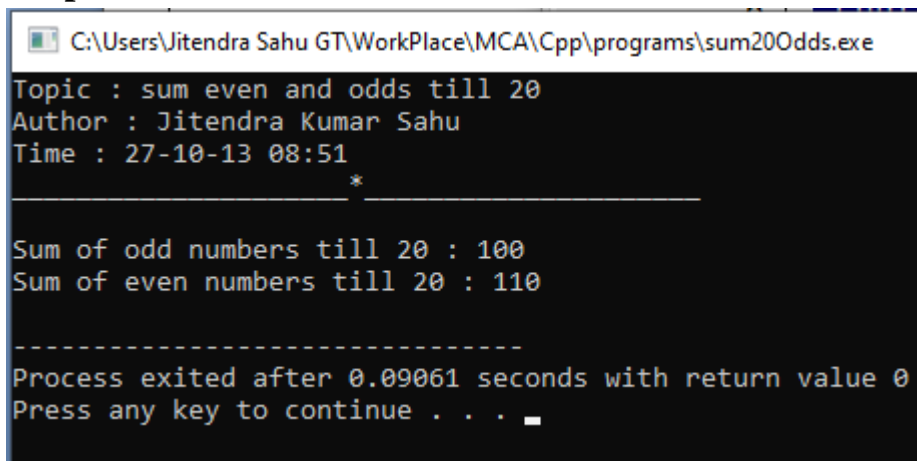
Output:

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)
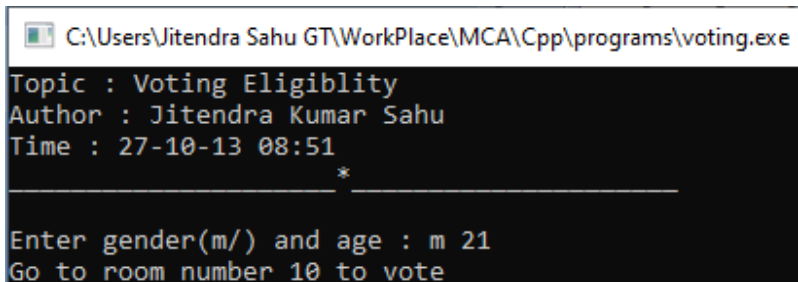
Code :

```cpp
#include<iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____\n"<< endl ;
}
int main()
{
        printIntro("Voting Eligiblity","27-10-13 08:51") ;
        int age ;
        char gender ;
        cout << "Enter gender(m/) and age : " ;
        cin >> gender >> age ;

        if (age >= 18 ) {
                if(gender=='m' || gender=='M')
                        cout << "Go to room number 10 to vote" ;
                else if(gender=='f' || gender=='F')
                        cout << "Go to room number 12 to vote" ;
                else cout << "Go to room number 8 to vote" ;
        }else {
                cout << "Not Eligible for voting!" << endl ;
        }
        return 0;
}
```
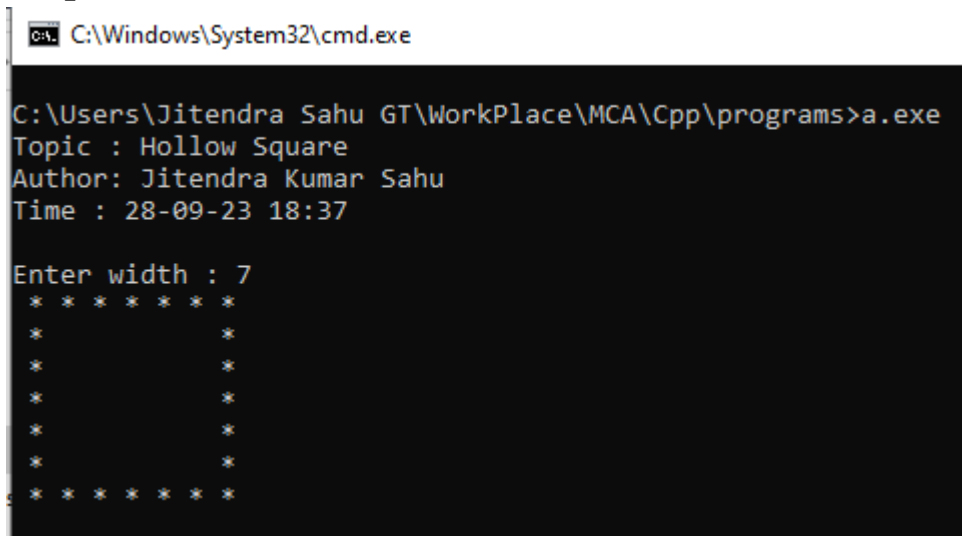
Output :

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

```cpp
#include <iostream>
#define area (length * width )
using namespace std ;

void printIntro(string topic, string time){
        cout <<"Topic : "<< topic<< endl ;
        cout << "Author: Jitendra Kumar Sahu" << endl ;
        cout <<"Time : "<< time << endl << endl;
}

void hollowSqure(int n){
        for(int i = 1 ; i <= n ; i++){
                for (int j = 1 ; j<= n ; j++)
                        if(j==1 || j == n || i==1 || i==n) cout << " *" ;
                        else cout << "  " ;
                cout << endl ;
        }
}

int main(){
        printIntro("Hollow Square", "28-09-23 18:37");
        int n ;
        cout << "Enter width : ";
        cin >> n ;
        hollowSqure(n);
        return 0 ;
}
```

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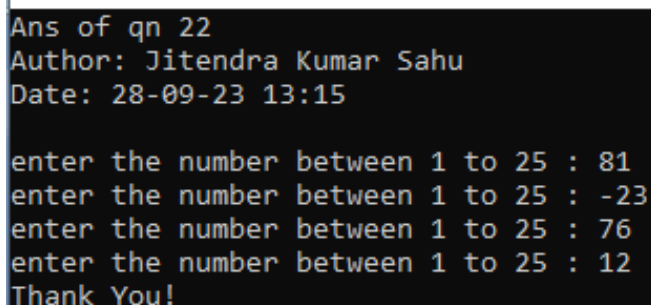
Program22: Write a C++ program to take input between 1-25 at runtime and display "Thank you user" for selecting number between 1-25 otherwise display "please enter number between 1-25 only!!".(using while loop).

Code :

```cpp
#include <iostream>
#include<ctime>
using namespace std;
int main() {
        cout<<"Shubham Kumar Singh "<<endl;
        cout<<"MCA- 1sem"<<endl;
        time_t now = time(0);
        char* dt = ctime(&now);
        cout<<"Date and Time is: "<<dt<<endl;
    int number;
    while (true) {
       cout << "Enter a number between 1 and 25: ";
       cin >> number;

       if (number >= 1 && number <= 25) {
          cout << "Well Done!" << endl;
          break; // Exit the loop if the input is valid
       } else {
          cout << "Please enter a number between 1 and 25 only!!" << endl;
       }
    }
    return 0;
}
```

Output :

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\1-to-25.exe

```
Ans of qn 22
Author: Jitendra Kumar Sahu
Date: 28-09-23 13:15

enter the number between 1 to 25 : 81
enter the number between 1 to 25 : -23
enter the number between 1 to 25 : 76
enter the number between 1 to 25 : 12
Thank You!

-------------------------------
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}

int main(){
        printIntro("","") ;

        int n = 0;
        do{
                cout << "Enter a number : " ;
                cin >> n ;
                if(n > 0 ) cout << "Positive" ;
                else if(n < 0 ) cout << "Negative" ;
                else cout << "Zero" ;
                cout << endl ;
        }while(n != 0) ;

        return 0 ;
}
```

Output:

Program 24: Write a C++ program to differentiate break and continue jump statement.
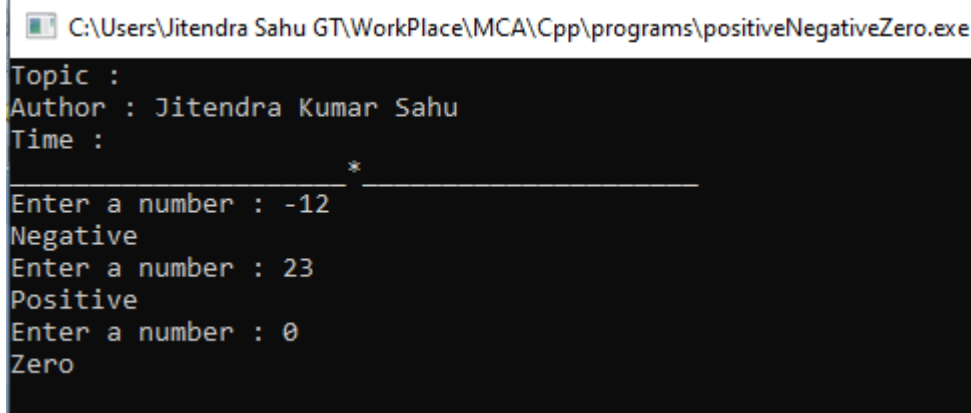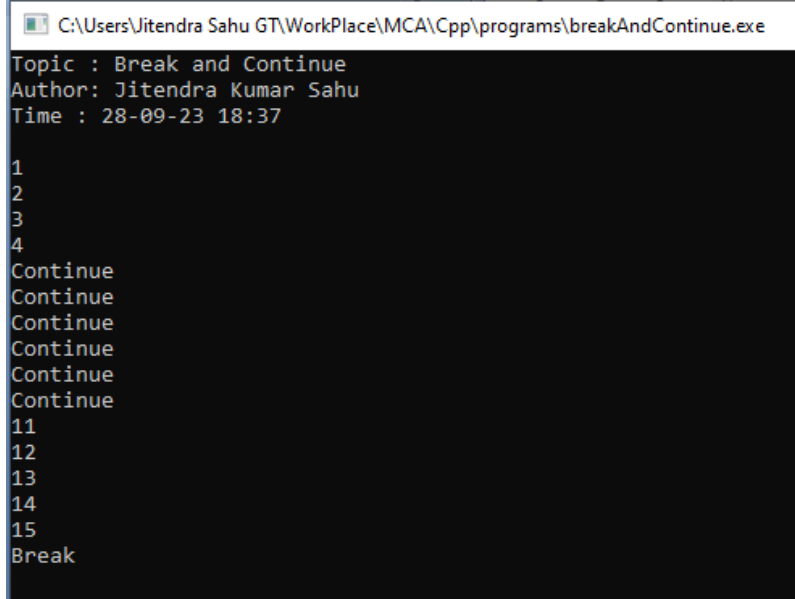
Code

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time){
cout <<"Topic : "<< topic<< endl ;
cout << "Author: Jitendra Kumar Sahu" << endl ;
cout <<"Time : "<< time << endl << endl;
}

int main(){
        printIntro("Break and Continue", "28-09-23 18:37");
        int n  = 20 ;
        for (int i = 1 ; i <= 20 ; i++){
                // continue statment will only skip some iterations
                if ( i > 4 && i <= 10 ) {
                    cout << "Continue" << endl ;
                    continue ;
                }
                // break will get the controll out of the loop
                if ( i == 16 ) {
                    cout << "Break" << endl ;
                    break ;
                }
                cout << i << endl ;
        }
        return 0 ;
}
```

Output:

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\breakAndContinue.exe

Topic : Break and Continue
Author: Jitendra Kumar Sahu
Time : 28-09-23 18:37

1
2
3
4
Continue
Continue
Continue
Continue
Continue
Continue
11
12
13
14
15
Break
```

Program 25. Write a C++ program to display details of 5 students' detail should be contained student name, roll number, marks using structure.

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string date) {
        cout << "Program topic : "<< topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Date : "<< date << endl ;
        cout <<"----------------*-----------------"<< endl ;
}

struct Student {
        int roll ;
        string name ;
        int marks ;
};

int main() {

        printIntro("Structure demonstration : storing details of 5 students' ","06-10-23 15:52") ;

        int n = 5 ;
        Student s[n] ;
        cout << "Enter student details \n" ;

        for(int i = 0 ; i< n ; i++ ) {
                cout << "Enter roll, name of student "<< (i+1) << " : "<< endl ;

                cin >> s[i].roll ;
                getline(cin,s[i].name) ; // this line is for handling dump
                getline(cin,s[i].name) ;
                cout << "Enter marks : " ;
                cin >> s[i].marks ;
        }

        for(int i = 0 ; i < n ; i++ ) {
                cout << endl << endl<< "Printing record of Student "<< (i+1) << endl ;
                cout << "Roll : "<< s[i].roll << endl ;
                cout << "Name : "<< s[i].name << endl ;
                cout << "Marks : " << s[i].marks << endl ;
        }
        return 0 ;
}
```

OUTPUT :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs>structStudent.exe
Program topic : Structure demonstration : storing details of 5 students'
Author : Jitendra Kumar Sahu
Date : 06-10-23 15:52
----------------*----------------
Enter student details
Enter roll, name of student 1 :
101
Jitendra Kumar
Enter marks : 83
Enter roll, name of student 2 :
102
Harishankar Ojha
Enter marks : 94
Enter roll, name of student 3 :
103
Shivam Desai
Enter marks : 85
Enter roll, name of student 4 :
104
Rameshwarm Teja
Enter marks : 75
Enter roll, name of student 5 :
105
Sittama Patle
Enter marks : 82


Printing record of Student 1
Roll : 101
Name : Jitendra Kumar
Marks : 83


Printing record of Student 2
Roll : 102
Name : Harishankar Ojha
Marks : 94


Printing record of Student 3
Roll : 103
Name : Shivam Desai
Marks : 85


Printing record of Student 4
Roll : 104
Name : Rameshwarm Teja
Marks : 75


Printing record of Student 5
Roll : 105
Name : Sittama Patle
Marks : 82
```

Program 26. Write a C++ program to create a structure named "Date" which contains three members Day, Month, Year and display current date entering by the user using function definition.

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string date) {
        cout << "Program topic : "<< topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Date : "<< date << endl ;
}

struct Date {
        int day, month, year ;
        void getDate() {
                cin >> day >> month >> year ;
        }
        void printDate() {
                cout << day<< "-" << month << '-' << year << endl ;
        }
};

int main() {

        printIntro("Date using structure","06-10-23 15:59") ;
        Date d = {6,10,2023} ;
        cout << "Enter date "<< endl ;
        d.getDate() ;
        d.printDate() ;
        return 0 ;

}
```
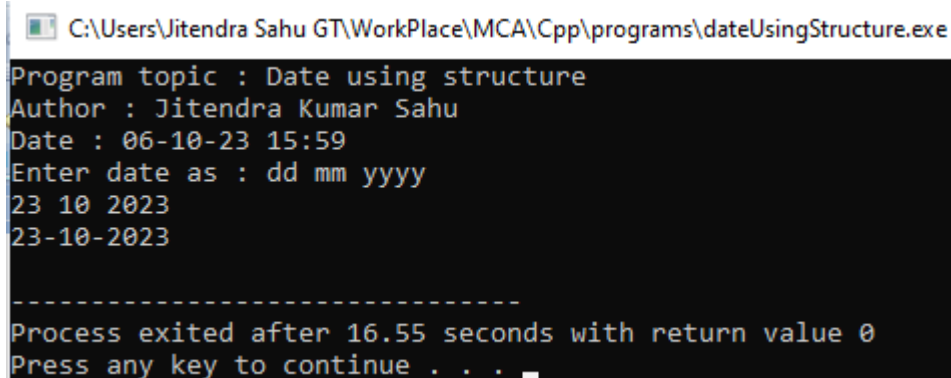
Output ;

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\dateUsingStructure.exe

Program topic : Date using structure
Author : Jitendra Kumar Sahu
Date : 06-10-23 15:59
Enter date as : dd mm yyyy
23 10 2023
23-10-2023

--------------------------------
Process exited after 16.55 seconds with return value 0
Press any key to continue . . .
```

Program : 27. Write a C++ program to demonstrate enum with switch case.

Code :

```cpp
#include <iostream>
using namespace std ;

// definition of enumuration
enum week{
        mon=1, tue, wed , thu, fri ,sat ,sun
};

void printIntro(string topic, string date) {
        cout << "Program topic : "<< topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Date : "<< date << endl ;
        cout << "------------*-----------" << endl ;
}

int main() {

  printIntro("week day with enum", "06-10-23 16:36") ;

        cout << "Enter an integer between 1 and 7 : ";

        int x ;
        cin >> x ;
        switch(x){
                case mon : cout << "Monday" << endl ;
                break ;

                case tue : cout << "Tueday" << endl ;
                break ;

                case wed : cout << "Wednessday" << endl ;
                break ;

                case thu : cout << "Thursday" << endl ;
                break ;

                case fri : cout << "Friday" << endl ;
                break ;

                case sat : cout << "Saturday" << endl ;
                break ;

                case sun : cout << "Sunday" << endl ;
                break ;
```
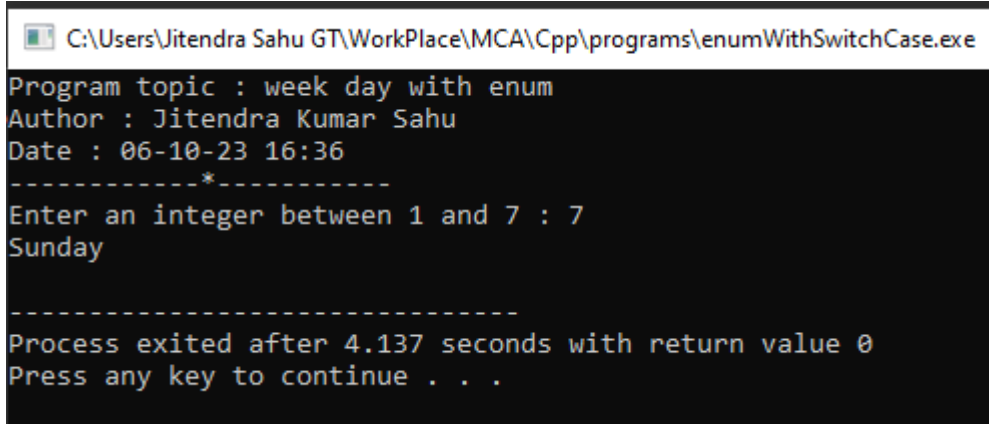
```
            default : cout << "Nahi, mai mere ek week me " << x << " din nahi hota" ;
    }
    return 0 ;
}
```

Output :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\enumWithSwitchCase.exe

Program topic : week day with enum
Author : Jitendra Kumar Sahu
Date : 06-10-23 16:36
-----------*----------
Enter an integer between 1 and 7 : 7
Sunday


---------------------------------
Process exited after 4.137 seconds with return value 0
Press any key to continue . . .
```

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 :

```cpp
#include<iostream>
using namespace std;

enum numbers{
        one = 1 , two, three , four,five, six , seven , eight , nine ,ten
};

void printIntro(string topic, string time){
        cout << "Program topic : " << topic<< endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout << "---------------*----------------"<< endl<< endl ;
}
int main()
{
        printIntro("printing numbers with enum","10-10-23 02:59") ;

        cout << one << " one"<< endl ;
        cout << two << " two" << endl ;
        cout << three << " three"<< endl ;
        cout << four << " four" << endl ;
        cout << five  << " five" << endl ;
        cout << six << " six" << endl ;
        cout << seven << " seven" << endl ;
        cout << eight << " eight" << endl ;
        cout << nine << " nine"<< endl ;
        cout << ten << " ten"  << endl ;
}
```
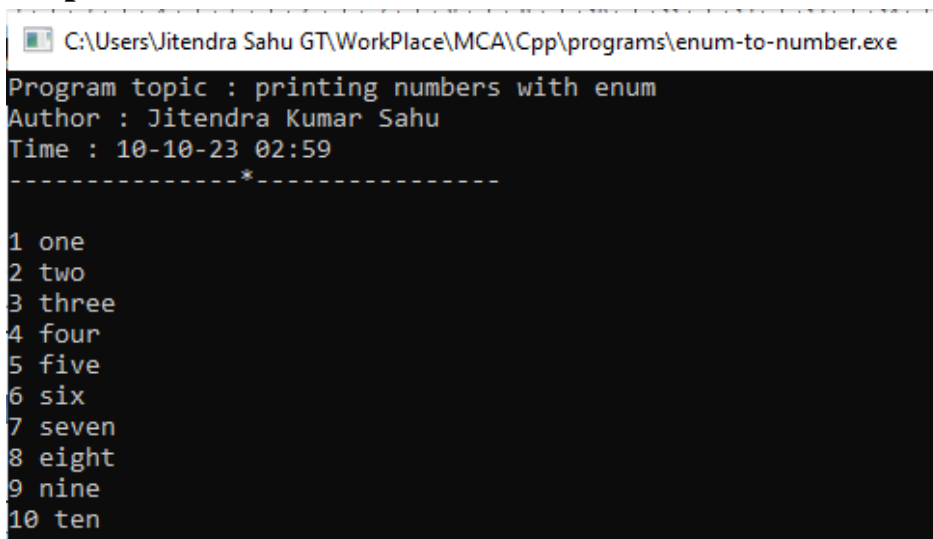
Output :

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

Code :

```cpp
#include<iostream>
using namespace std;

void printIntro(string topic, string time){
        cout << "Program topic : " << topic<< endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout << "---------------*----------------"<< endl<< endl ;
}


void swap(){
        int a = 5, b = 6 ;
        cout << "a " << a << endl ;
        cout << "b " << b << endl ;

        a = a + b ;
        b = a - b ;
        a = a - b ;

        cout << "\nafter swap : " << endl ;
        cout << "a " << a << endl ;
        cout << "b " << b << endl ;
}

int main()
{
        printIntro("Swap number without using thired variable","10-10-23 03:18");
        swap();
        return 0 ;
}
```
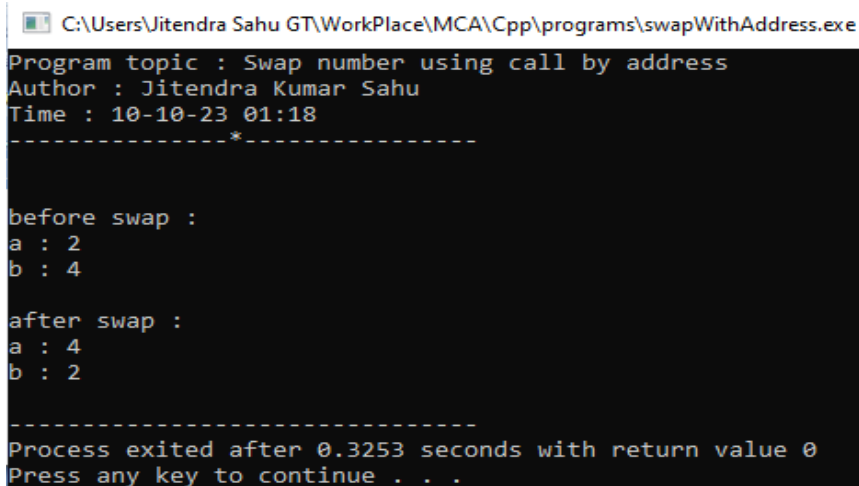
Output :

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

Code :

```cpp
#include<iostream>
using namespace std;
void printIntro(string topic, string time){
        cout << "Program topic : " << topic<< endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout << "---------------*----------------"<< endl<< endl ;
}
void swap(int *a, int *b){
        int temp = *a;
        *a =  *b;
        *b = temp ;
        cout << "\nafter swap : " << endl ;
        cout << "a : " << *a << endl ;
        cout << "b : " << *b << endl ;
}

int main()
{
        printIntro("Swap number using call by address ","10-10-23 01:18");
        int a = 2 , b = 4 ;

        cout << "\nbefore swap : " << endl ;
        cout << "a : " << a << endl ;
        cout << "b : " << b << endl ;
        swap(&a, &b);
        return 0 ;

}
```
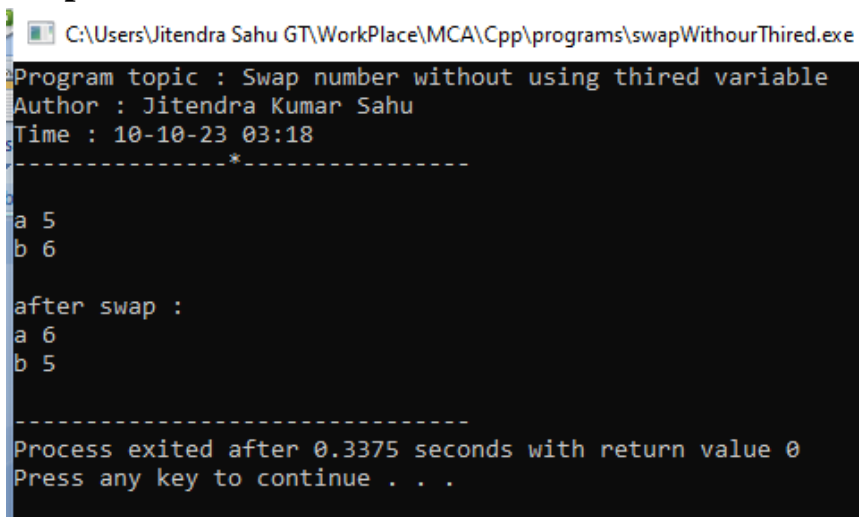
Output :

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}
void change(int a){
        //changing value
        a = a+1 ;
        cout<< "updated value at function : "<< a <<endl ;
}
int main(){
        printIntro("Illustration of call by value","28-10-23 19:08") ;
        int x = 5 ;
        cout << "before call " << x << endl ;
        change(x) ;
        cout << "after call " << x << endl ;
        return 0 ;
}
```

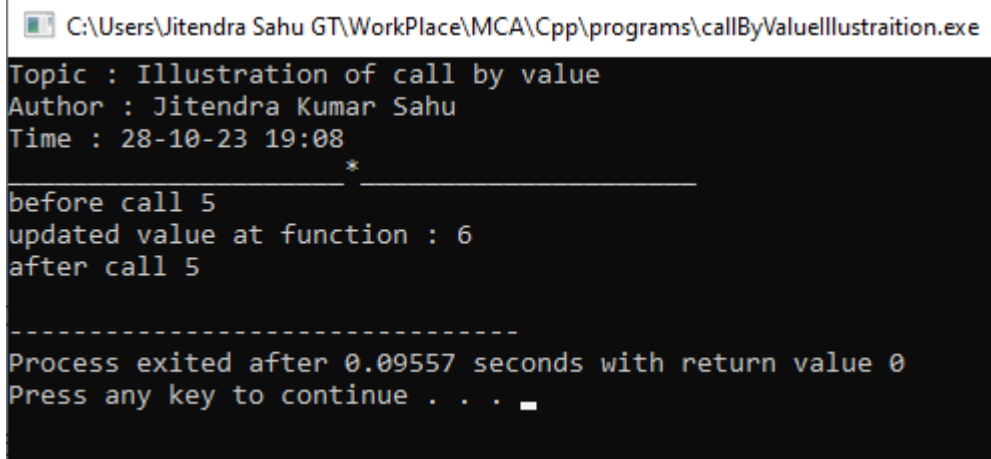**Output :**
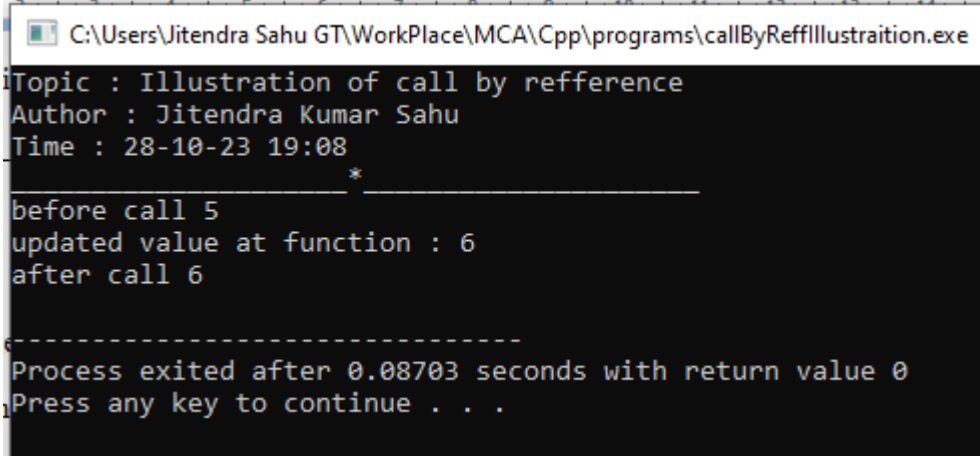
Program 32. Write a C++ program to illustrate working of call by reference method of a function invoking.

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
cout<<"Topic : " << topic << endl ;
cout<<"Author : Jitendra Kumar Sahu" << endl ;
cout << "Time : "<< time << endl ;
cout <<"_____*_____"<< endl ;
}

void change(int &a){
        //changing value
        a = a+1 ;
        cout<< "updated value at function : "<< a <<endl ;
}

int main(){
        printIntro("Illustration of call by refference","28-10-23 19:08") ;
        int x = 5 ;
        cout << "before call " << x << endl ;
        change(x) ;
        cout << "after call " << x << endl ;
        return 0 ;
}
```

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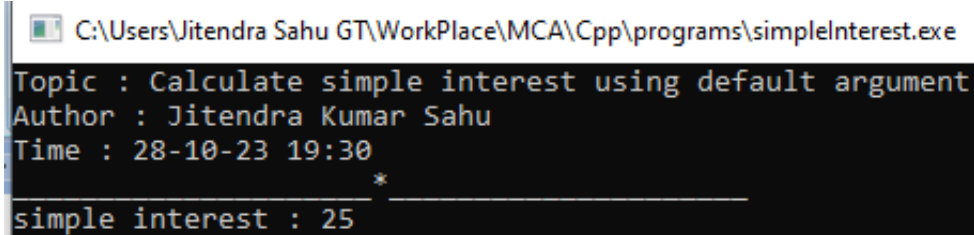
Program 33. Write a C++ program to calculate simple interest using default arguments.

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}
float getSimpleInterest(float p=0,float r=0 , float t=0){
        return (p*r*t)/100;
}
int main(){
        printIntro("Calculate simple interest using default argument","28-10-23 19:30") ;
        float p = 100, r = 5 , t = 5 ;
        cout << "simple interest : " << getSimpleInterest(p,r,t);
        cout << endl ;
        return 0 ;
}
```

Output:


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\simpleInterest.exe

Topic : Calculate simple interest using default argument
Author : Jitendra Kumar Sahu
Time : 28-10-23 19:30
                        *
simple interest : 25
```

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

Code :
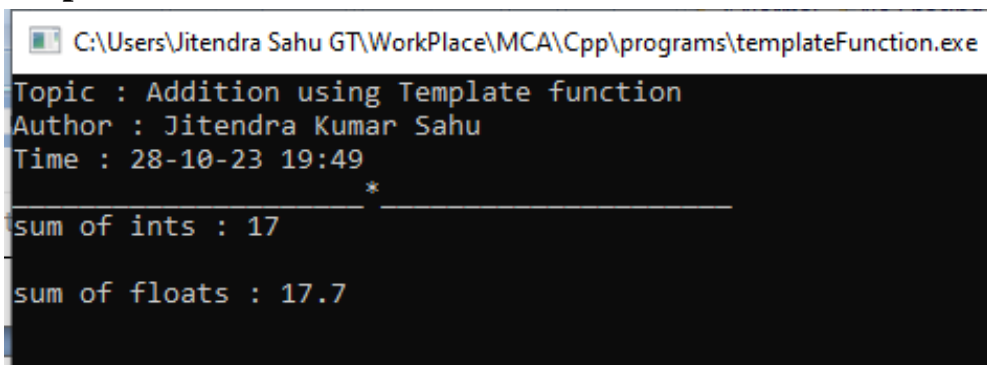
```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}

template <typename T>
void add(T a, T b){
        cout << a+b << endl ;
}

int main(){
        printIntro("Addition using Template function","28-10-23 19:49") ;

        cout << "sum of ints : " ;
        add(6,11) ;
        cout << endl ;

        cout << "sum of floats : " ;
        add(6.5,11.2) ;
        cout << endl ;

        return 0 ;
}
```

Output :

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

Code :

```cpp
#include <iostream>
using namespace std;

void printIntro(string topic, string time)
{
    cout << "Topic : " << topic << endl;
    cout << "Author : Jitendra Kumar Sahu" << endl;
    cout << "Time : " << time << endl;
    cout << "_____*_____" << endl;
}


template <typename T, typename U>
class Calc
{

public:
    void operate(T a, T b, char operation)
    {
        switch (operation)
        {
        case '+':
            cout << a + b << endl;
            break;

        case '-':
            cout << a - b << endl;
            break;

        case '*':
            cout << a * b << endl;
            break;

        case '/':
            cout << a / b << endl;
            break;

        default:
            cout << "Operation not defined " << endl;
        }
    }
};
```

```cpp
int main()
{

        printIntro("calculator using template class", "28-10-23 19:57");
        Calc<int, int> c;
        Calc<float, float> d;

        int a = 5, b = 6;
        float x = 5.2f, y = 5.3f;

        cout << a << " + " <<b << " : "  ;
        c.operate(a, b, '+');

        cout << a << " - " <<b << " : "  ;
        c.operate(a, b, '-');

        cout << a << " * " <<b << " : "  ;
        c.operate(a, b, '*');

        cout << a << " / " <<b << " : "  ;
        c.operate(a, b, '/');

        cout << x << " + " << y << " : "  ;
        d.operate(x, y, '+');

        cout << x << " - " << y << " : "  ;
        d.operate(x, y, '-');

        cout << x << " * " << y << " : "  ;
        d.operate(x, y, '*');

        cout << x << " / " << y << " : "  ;
        d.operate(x, y, '/');

        return 0;
}
```

Output:

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\calculatorWithTemplate.exe

Topic : calculator using template class
Author : Jitendra Kumar Sahu
Time : 28-10-23 19:57
_____ * _____
5 + 6 : 11
5 - 6 : -1
5 * 6 : 30
5 / 6 : 0
5.2 + 5.3 : 10.5
5.2 - 5.3 : -0.1
5.2 * 5.3 : 27.56
5.2 / 5.3 : 0.981132
```

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

Code:

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string date) {
        cout << "Program topic : "<< topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Date : "<< date << endl ;
        cout << "_____*_____"<<endl << endl ;


}


inline float area(int r){
        const float pi = 22/7 ;
        return pi*r*r ;
}
int main() {

        printIntro("inline function to calculate simple interest","02-11-23 15:34") ;
        int  radius = 5 ;
        cout << "area of circle : " << area(radius) ;
        return 0 ;

}
```

Code ::


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\inlineAreaOfCircle.exe

Program topic : inline function to calculate simple interest
Author : Jitendra Kumar Sahu
Date : 02-11-23 15:34
_____*_____

area of circle : 75
------------------------------
Process exited after 0.08239 seconds with return value 0
Press any key to continue . . .
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string date) {
        cout << "Program topic : "<< topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Date : "<< date << endl ;
        cout << "_____*_____"<<endl << endl ;
}

int add(int a, int b){
        return a+b ;
}

int add(int a, int b , int c){
        return a+b+c ;
}

int main() {

        printIntro("function overloading","02-11-23 15:34") ;

        cout << "Addition of 2 integers : " << add(2,3) << endl ;
        cout << "Addition of 3 integers : " << add(2,3,4) << endl ;
        return 0 ;
}
```
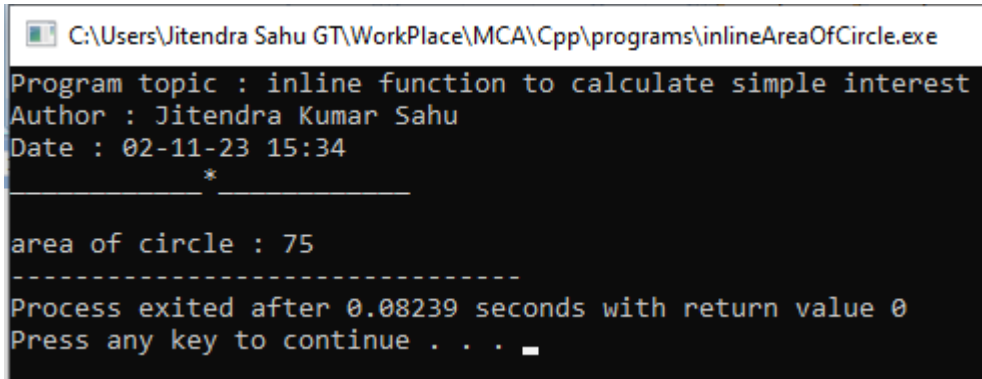
Output :


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\functionOverloading.exe

Program topic : function overloading
Author : Jitendra Kumar Sahu
Date : 02-11-23 15:34
_____ * _____

Addition of 2 integers : 5
Addition of 3 integers : 9
```

Program 38. Write a C++ program to find the size of 1-D, 2-D and multidimensional array.

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}

int main(){

        printIntro("size of different types of array","02-11-23 18:16") ;

        int p, r, c ;

        cout << "\n1D array \n" ;
        cout << "enter number of elements : " ;
        cin >> c ;
        int arr1d[c] ; // 2d array
        cout << "number of elements : " << c << endl ;
        cout << "size : " << sizeof(arr1d) << endl ;

        cout << "\n2D array \n" ;
        cout << "enter number of rows and columns : " ;
        cin >> r >> c ;
        int arr2d[r][c] ; // 2d array
        cout << "number of elements : " << (r*c)<< endl ;
        cout << "size : " << sizeof(arr2d) << endl ;

        cout << "\n3D array \n" ;
        cout << "enter number of pages, rows and columns : " ;
        cin >> p >> r >> c ;
        int arr3d[p][r][c] ; // 3d array
        cout << "number of elements : " << (p*r*c)<< endl ;
        cout << "size : " << sizeof(arr3d) << endl ;
        return 0 ;
}
```
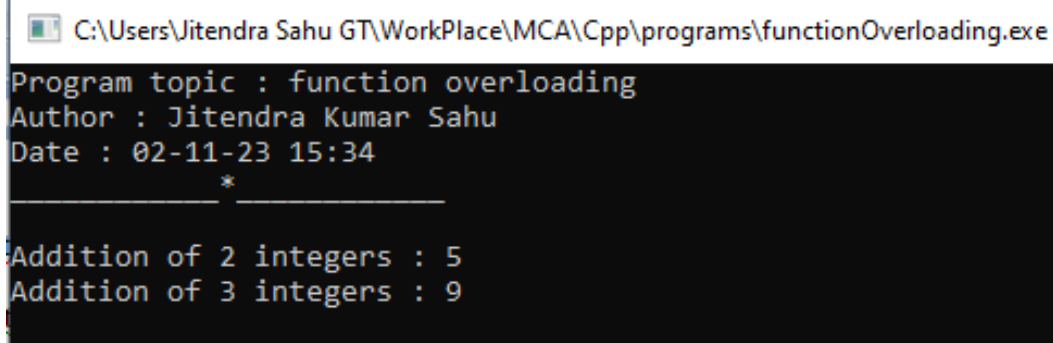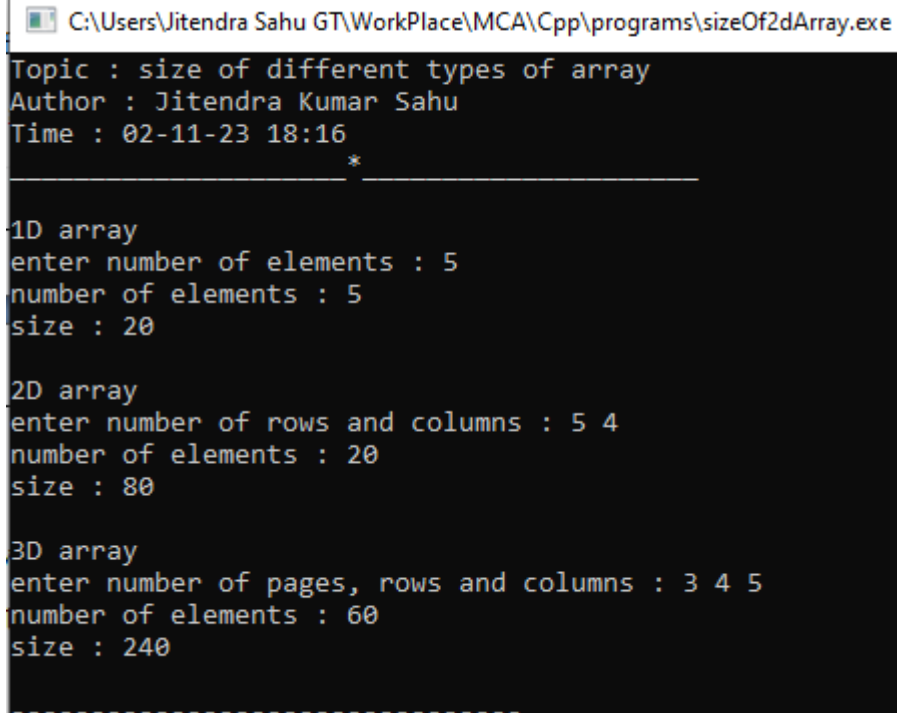
Output :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\sizeOf2dArray.exe
Topic : size of different types of array
Author : Jitendra Kumar Sahu
Time : 02-11-23 18:16
                        *
_____  _____

1D array
enter number of elements : 5
number of elements : 5
size : 20

2D array
enter number of rows and columns : 5 4
number of elements : 20
size : 80

3D array
enter number of pages, rows and columns : 3 4 5
number of elements : 60
size : 240

----------------------------------
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}

int main(){

        printIntro("avarage of array elements","02-11-23 18:08") ;

        int n = 7, sum = 0 ;
        int arr[n] = {12,13,14,15,16,17,18} ;

        cout <<"Array elements : " ;
        for(int i = 0 ; i < n ; i++ ){
                        cout << arr[i] << " ";
                        sum += arr[i] ;
        }
        cout << endl ;
        cout << "average : " << (sum/n) << endl ;

        return 0 ;
}
```

Output :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\averageOfArrayElements.exe

Topic : avarage of array elements
Author : Jitendra Kumar Sahu
Time : 02-11-23 18:08
                    *_____
Array elements : 12 13 14 15 16 17 18
average : 15
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}

int main(){
        printIntro("Print Reversed Array","02-11-23 18:00") ;
        int n = 5 ;
        int arr[n] = {11,12,13,14,15}  ;

        cout << "Our array : " << endl ;
        for (int i = 0 ; i < n ; i++ ){
                cout << arr[i] << " " ;
        }
        cout << endl ;
        cout << "in reversed order : " << endl ;

        for (int i = n-1 ; i >= 0 ; i--){
                cout << arr[i] << " " ;
        }
        cout << endl ;
        return 0 ;
}
```
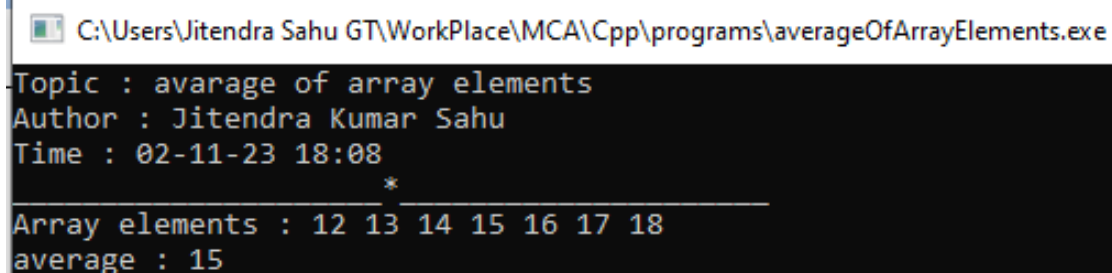
Output :


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\reversedArrayPrint.exe

Topic : Print Reversed Array
Author : Jitendra Kumar Sahu
Time : 02-11-23 18:00
_____*_____
Our array :
11 12 13 14 15
in reversed order :
15 14 13 12 11
```

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 :

```cpp
#include <iostream>
using namespace std;

void printIntro(string topic, string time)
{
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

class Student
{
        int roll, marks[5];
        string name;

public:
        void getData();
        void showData();
        void totalMarks();
};

void Student ::getData()
{
        cout << "enter roll , name : ";
        cin >> roll;
        getline(cin, name);
        getline(cin, name);
        cout << "enter five marks : ";
        for (int i = 0; i < 5; i++){
                cin >> marks[i];
        }
}
```
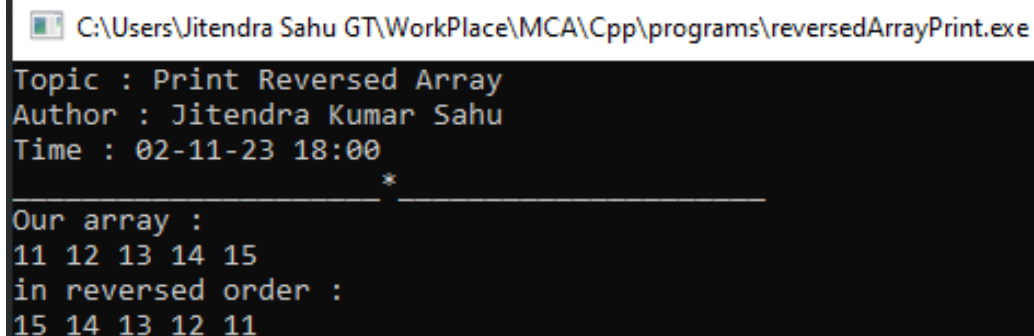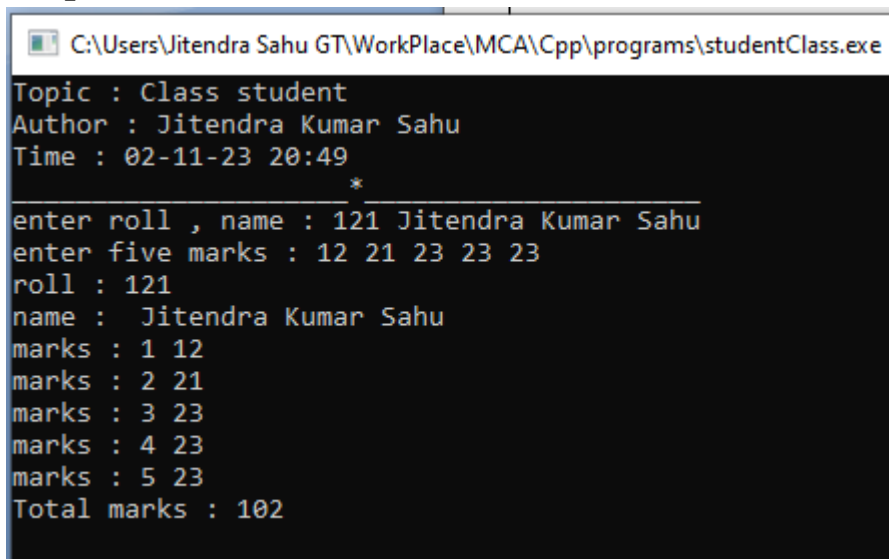
```cpp
void Student ::showData()
{
        cout << "roll : " << roll << endl;
        cout << "name : " << name << endl;
        for (int i = 0; i < 5; i++){
                cout << "marks : " << (i + 1) << " " << marks[i] << endl;
        }
}
void Student ::totalMarks()
{
        int sum = 0;
        for (int i = 0; i < 5; i++)
        {
                sum += marks[i];
        }
        cout << "Total marks : " << sum << endl;
}

int main()
{
        printIntro("Class student", "02-11-23 20:49");
        Student jitendra;
        jitendra.getData();
        jitendra.showData();
        jitendra.totalMarks();
        return 0;
}
```

Output :


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\studentClass.exe

Topic : Class student
Author : Jitendra Kumar Sahu
Time : 02-11-23 20:49
                        *_____
enter roll , name : 121 Jitendra Kumar Sahu
enter five marks : 12 21 23 23 23
roll : 121
name :  Jitendra Kumar Sahu
marks : 1 12
marks : 2 21
marks : 3 23
marks : 4 23
marks : 5 23
Total marks : 102
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}

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(){

      printIntro("Nesting of member function","02-11-23 22:33") ;
      ThreeNum t ;
      t.a = 43 ;
      t.b = 45 ;
      t.c = 63 ;

      t.display() ;
      t.displayMax() ;

      return 0 ;
}
```

Output :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\maxOf3NestedFunc.exe

Topic : Nesting of member function
Author : Jitendra Kumar Sahu
Time : 02-11-23 22:33
_____*_____
a : 43, b : 45, c : 63
max : 63

------------------------------------------------
```

Program 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 :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
cout<<"Topic : " << topic << endl ;
cout<<"Author : Jitendra Kumar Sahu" << endl ;
cout << "Time : "<< time << endl ;
cout <<"_____*_____"<< endl ;
}

class Calculator{
        int a = 5 , b = 4 ;
        public :
        void add(){
                cout << "sum : " << a+b << endl;
        }
        void sub(){
                cout << "sub : " << a-b  << endl ;
        }
        void mult(){
                cout << "mult : " << a*b  << endl;
        }
        void div(){
                cout << "div : " << a/b  << endl ;
        }
};

int main(){
        printIntro("class as calculator","02-11-23 22:42") ;
        Calculator c ;
        c.add() ;
        c.sub() ;
        c.mult() ;
        c.div() ;
        return 0 ;
}
```
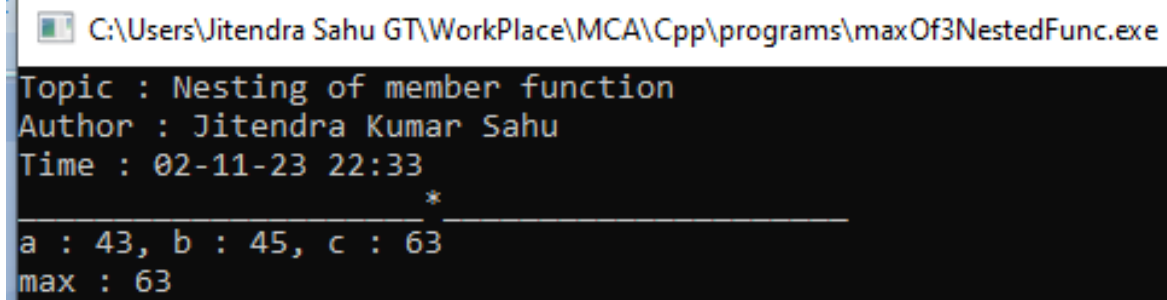
Output :

Topic : class as calculator
Author : Jitendra Kumar Sahu
Time : 02-11-23 22:42
                              *
sum : 9
sub : 1
mult : 20
div : 1

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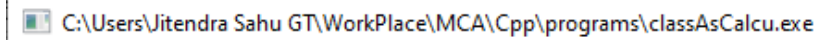
Code :

```cpp
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}

inline int inlineSqure(int x){
        return x*x ;
}

int main(){

        printIntro("Outside function inline","02-11-23 23:41") ;
        int k = 5 ;
        cout << k << " squared = " << inlineSqure(k) << endl ;
        return 0 ;
}
```

Output :

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\program

Topic : Outside function inline
Author : Jitendra Kumar Sahu
Time : 02-11-23 23:41
                        *
5 squared = 25


--------------------------------
Process exited after 0.08455 seconds with
```

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

Code :
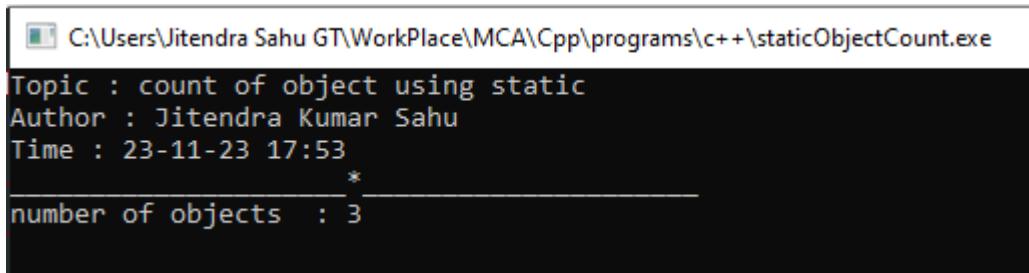
```cpp
#include <iostream>
using namespace std ;

class A{
        static int numberOfObjects ;
        public :
        // a static function that uses static variable
        static int getNumberOfObjects(){
                return numberOfObjects ; // returning static variable
        }
        void admit(){
                numberOfObjects++ ;
        }
} ;
int A :: numberOfObjects = 0 ;
void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}
int main(){
        printIntro("count of object using static","23-11-23 17:53") ;

        A object1 , object2 , object3; // creating object
        // calling function that work on static variable
        object1.admit() ;
        object2.admit() ;
        object3.admit() ;
        cout << "number of objects  : " ;
        cout << A::getNumberOfObjects() << endl ;
        return 0 ;
}
```
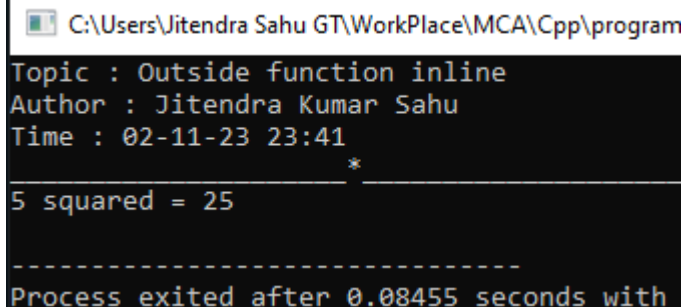
Ouput :

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 :

```cpp
#include <iostream>
using namespace std ;

class Student{
    string name, remarks ;
    int rno, score ;
    void assignRem() ;
    public :
         void enter() ;
         void display() ;
} ;
void Student :: assignRem(){
    score >= 50 ? remarks = "selected" : remarks = "not selected" ;
}
void Student :: enter(){
    cout << "Enter registration number : " ;
    cin >> rno ;
    cout << "Enter name : " ;
    getline(cin,name) ;
    getline(cin,name) ;
    cout << "Enter score : " ;
    cin >> score ;
    assignRem() ;
}
void Student :: display(){
  cout << "------------------------\n";
  cout << "registration no : " << rno << endl ;
  cout << "name : " << name << endl ;
```

```cpp
    cout << "score : " << score << endl ;
    cout << "remarks : " << remarks << endl ;
  }
  void printIntro(string topic, string time) {
         cout<<"Topic : " << topic << endl ;
         cout<<"Author : Jitendra Kumar Sahu" << endl ;
         cout << "Time : "<< time << endl ;
         cout <<"_____*_____"<< endl ;
  }
  int main(){
         printIntro("Nesting of member function","23-11-23 17:53") ;

         Student jitendra ;
         jitendra.enter() ;
         jitendra.display() ;
         return 0 ;
  }
```

Ouput :

C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\c++\studentClass.exe

```
Topic : Nesting of member function
Author : Jitendra Kumar Sahu
Time : 23-11-23 17:53
                           *
Enter registration number : 1001
Enter name : Jitendra Kumar Sahu
Enter score : 81
------------------------
registration no : 1001
name : Jitendra Kumar Sahu
score : 81
remarks : selected
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

class Parent{
       public :
        int parentDataMember ;
} ;

class Child : public Parent{

       public :
        int childDataMember ;

        void setParentDataMember(int x){
               parentDataMember = x ;
        }
        void setChildDataMember(int x){
               childDataMember = x ;
        }
        void printValues(){
               cout << "Child Data member value : "<< childDataMember << endl ;
               cout << "Parent Data member value : "<< parentDataMember << endl ;
        }
} ;

void printIntro(string topic, string time) {
       cout<<"Topic : " << topic << endl ;
       cout<<"Author : Jitendra Kumar Sahu" << endl ;
       cout << "Time : "<< time << endl ;
       cout <<"_____*_____"<< endl ;
}

int main(){
       printIntro("Single inheritence","23-11-23 17:53") ;

       Child object ;
       object.setParentDataMember(1) ;
       object.setChildDataMember(2) ;
       object.printValues() ;
       return 0 ;
}
```

Output :


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\c++\singleInheritence.exe

Topic : Single inheritence
Author : Jitendra Kumar Sahu
Time : 23-11-23 17:53
1_____*_____
Child Data member value : 2
Parent Data member value : 1
```

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

```cpp
#include <iostream>
using namespace std ;

class Father{
      public :
       void cookFood(){
        cout << "I am cooking rice!"<< endl ;
       }
} ;
class Mother{
      public :
       void goToOffice(){
             cout << "I am going to office"<<endl ;
       }
} ;

class Child : public Father , public Mother{
      public :
      void eat(){
             cout << "I am eating rice" << endl ;
      }
} ;

void printIntro(string topic, string time) {
      cout<<"Topic : " << topic << endl ;
      cout<<"Author : Jitendra Kumar Sahu" << endl ;
      cout << "Time : "<< time << endl ;
      cout <<"_____*_____"<< endl ;
}

int main(){
      printIntro("Multiple Inheritence","23-11-23 17:53") ;

      Child jitendra ;
      jitendra.cookFood() ;
      jitendra.eat() ;
      jitendra.goToOffice() ;
      return 0 ;
}
```
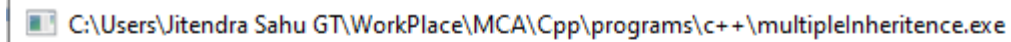
Output :


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\c++\multipleInheritence.exe

Topic : Multiple Inheritence
Author : Jitendra Kumar Sahu
Time : 23-11-23 17:53
_____ * _____
I am cooking rice!
I am eating rice
I am going to office
```

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

Code :

```cpp
#include <iostream>
using namespace std ;
class Vhicle{
        public :
         void start(){
           cout << "Vhicle is starting!"<< endl ;
        }
} ;
class Car: public Vhicle{
        public :
         void run(){
                cout <<"Car is running!"<<endl ;
        }
} ;
class Creta : public Car{
        public :
        void accelerate(){
                cout << "accelerating the speed" << endl ;
        }
} ;
void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}
int main(){
        printIntro("Multiple Inheritence","23-11-23 17:53") ;
        Creta yourCreta ;
        yourCreta.start() ;
        yourCreta.run() ;
        yourCreta.accelerate() ;
        return 0 ;
}
```
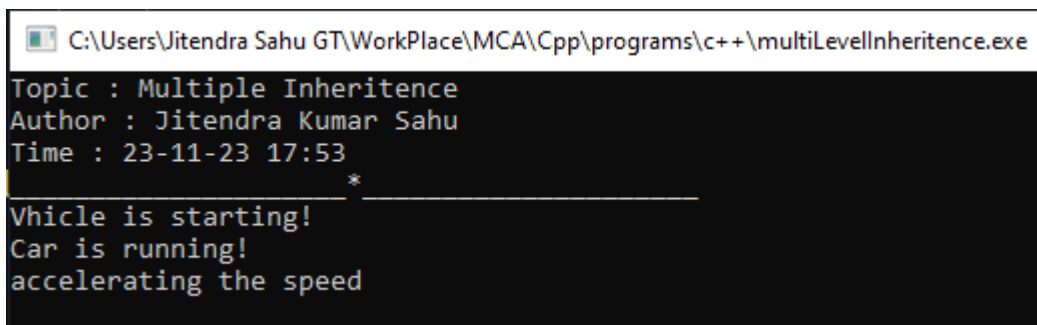
Output :


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\c++\multiLevelInheritence.exe

Topic : Multiple Inheritence
Author : Jitendra Kumar Sahu
Time : 23-11-23 17:53
_____*_____
Vhicle is starting!
Car is running!
accelerating the speed
```

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

Code :

```cpp
#include <iostream>
using namespace std;

class ParamConstruct
{
private:
        // Three private members
        int num1, num2, num3;
public:
        // constructor declaration
        ParamConstruct(int num1, int num2, int num3);
        void printData();
};
// constructor definition
ParamConstruct::ParamConstruct(int num1, int num2, int num3){
        this->num1 = num1;
        this->num2 = num2;
        this->num3 = num3;
}

void ParamConstruct ::printData(){
        cout << "num1 : " << num1 << endl;
        cout << "num2 : " << num2 << endl;
        cout << "num3 : " << num3 << endl;
}

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "------------------*------------------" << endl;
}

int main(){
        printIntro("Parametrized constructor demonstration", "24-11-23 22:46");

        int num1, num2, num3;
        cout << "Enter three integers : " ;
        cin >> num1 >> num2 >> num3 ;

        // creating object while passing values in to constructor
        ParamConstruct obj(num1, num2, num3);
        cout << "printing the members of object...\n";
```
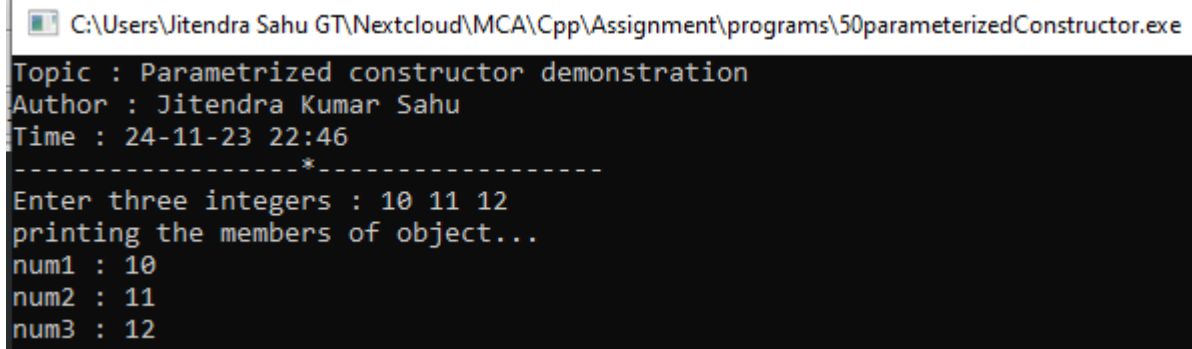
```
        obj.printData();
        return 0;
}
```

Output :


```
C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\50parameterizedConstructor.exe

Topic : Parametrized constructor demonstration
Author : Jitendra Kumar Sahu
Time : 24-11-23 22:46
-------------------*-------------------
Enter three integers : 10 11 12
printing the members of object...
num1 : 10
num2 : 11
num3 : 12
```

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 :

```cpp
#include <iostream>
using namespace std ;

class Sub{
      int ppr1_code , ppr2_code ;
      public :
  Sub(){
      ppr1_code = 101 ;
      ppr2_code = 102 ;
      cout << "ppr1_code : " << ppr1_code << endl ;
      cout << "ppr2_code : " << ppr2_code << endl ;
      cout << "internal object constructor called!\n" ;
  }
};
```
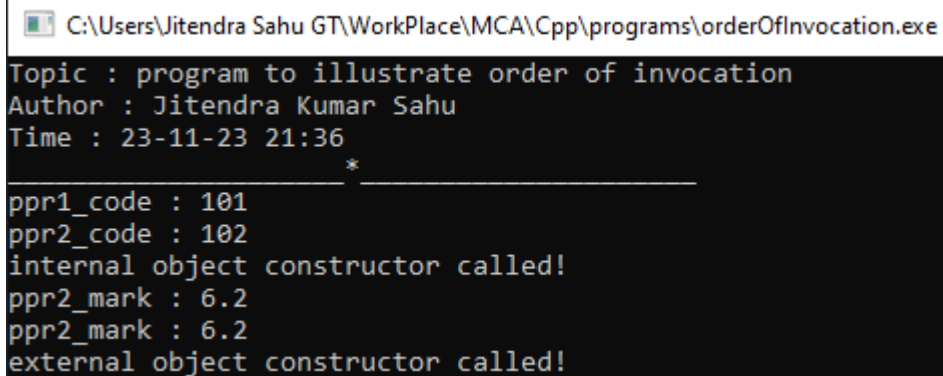
```cpp
class Mark{
        float ppr1_mark , ppr2_mark ;
        public :
   Mark(float a, float b){
        ppr1_mark = a ;
        ppr2_mark = b ;
        cout << "ppr2_mark : " << ppr2_mark << endl ;
        cout << "ppr2_mark : " << ppr2_mark << endl ;
        cout << "external object constructor called!\n" ;
   }
    Sub temp  ;
};

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"_____*_____"<< endl ;
}

int main(){
        printIntro("program to illustrate order of invocation","23-11-23 21:36") ;
        Mark object(5.1,6.2) ;
        return 0 ;
}
```

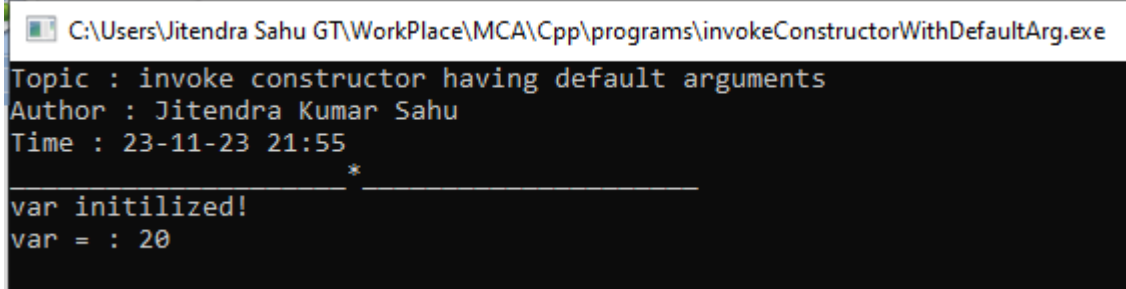**Output :**

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

Code

```cpp
#include <iostream>
using namespace std ;

class Test{
        int var ;
        public :
        Test(int a = 20){
                var = a ;
                cout << "var initilized! "<< endl ;
                cout << "var = : " << var << endl ;
        }
};

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____ *_____"<< endl ;
}

int main(){
        printIntro("invoke constructor having default arguments","23-11-23 21:55") ;
        Test object ;
        return 0 ;
}
```

Output


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\invokeConstructorWithDefaultArg.exe

Topic : invoke constructor having default arguments
Author : Jitendra Kumar Sahu
Time : 23-11-23 21:55
_____ *_____
var initilized!
var = : 20
```

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

Code

```cpp
#include <iostream>
using namespace std ;

class Test {
        int a ;
        int b ;

        public :
         Test(int x , int y){
                a = x ;
                b = y ;
         }
         // copy constructor
         Test(const Test &object){
                a = object.a ;
                b = object.b ;
         }
         void getData(){
                cout << "a : " << a << endl ;
                cout << "b : " << b << endl ;
         }
};

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}

int main(){
        Test t(22,55)   ;
        Test p(t) ;

        cout <<"values of object 1 \n" ;
        t.getData() ;

        cout<<"\nvalues of object 2 \n" ;
        p.getData() ;
        return 0 ;
}
```
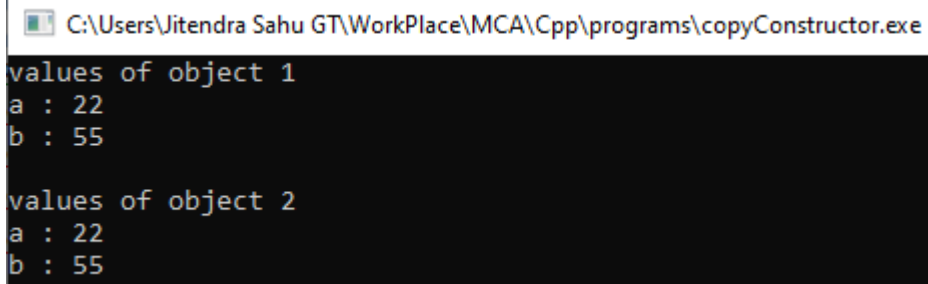
Output

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\copyConstructor.exe

values of object 1
a : 22
b : 55

values of object 2
a : 22
b : 55
```

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

Code

```cpp
#include <iostream>
using namespace std ;

class Scooter{
        string brandName ;
        string color ;

        public :
         Scooter(){
                brandName = "Honda" ;
                color = "white" ;
        }
         Scooter(string clr){
           brandName = "Suzuki"  ;
                color = clr ;
        }
         Scooter(string var, string clr){
                brandName = var ;
                color = clr ;
        }
         void printDetails(){
                cout << "Brand Name : " << brandName << endl ;
                cout << "color : " << color <<endl<< endl  ;
        }
 } ;

 void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"_____*_____"<< endl ;
}
int main(){
        printIntro("Constructor overloading","23-11-23 22:14") ;
        Scooter activa ;
        Scooter pleasure("red") ;
        Scooter mastro("hero","blue") ;
        activa.printDetails() ;
        pleasure.printDetails() ;
        mastro.printDetails() ;

        return 0 ;
}
```

Output


```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\ConstructorOverloading.exe

Topic : Constructor overloading
Author : Jitendra Kumar Sahu
Time : 23-11-23 22:14
_____ * _____
Brand Name : Honda
color : white

Brand Name : Suzuki
color : red

Brand Name : hero
color : blue
```

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

Code
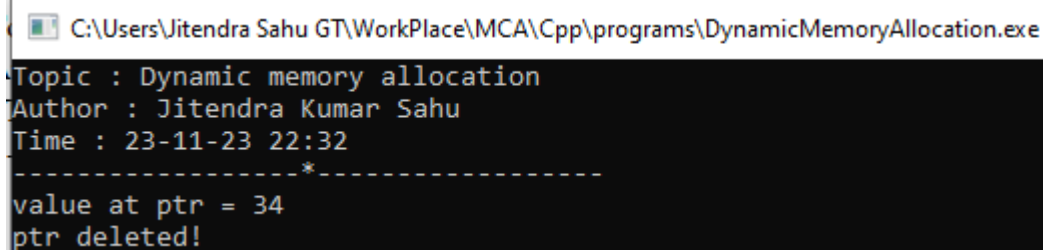
```
#include <iostream>
using namespace std ;

void printIntro(string topic, string time) {
        cout<<"Topic : " << topic << endl ;
        cout<<"Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : "<< time << endl ;
        cout <<"-----------------*------------------"<< endl ;
}

int main(){
        printIntro("Dynamic memory allocation","23-11-23 22:32") ;

        int *ptr = new int() ;
        if (ptr == NULL )
                cout << "unable to allocate memory!" ;
        else{
                *ptr = 34 ;
                cout <<"value at ptr = " << *ptr << endl ;
                delete ptr ;
                cout << "ptr deleted!" ;
        }
        return 0 ;
}
```

Output

```
C:\Users\Jitendra Sahu GT\WorkPlace\MCA\Cpp\programs\DynamicMemoryAllocation.exe

Topic : Dynamic memory allocation
Author : Jitendra Kumar Sahu
Time : 23-11-23 22:32
-----------------*----------------
value at ptr = 34
ptr deleted!
```

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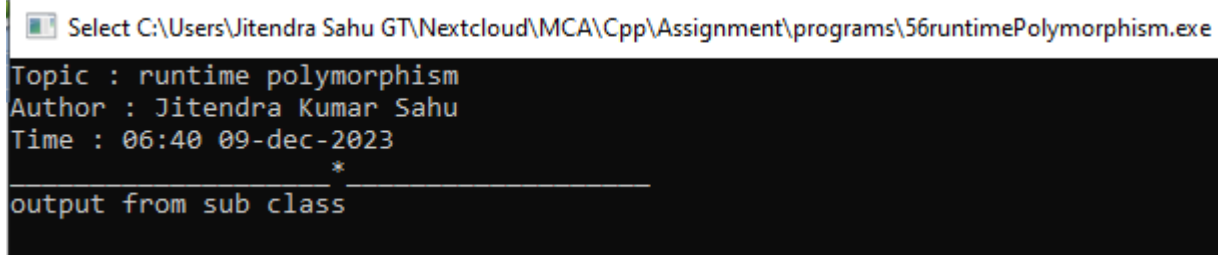
Code :

```
#include <iostream>
using namespace std ;

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : " << time << endl ;
        cout << "_____*_____" << endl ;
}

class Super{
        public :
                void print(){
                        cout << "output from super class" << endl ;
                }
} ;
class Sub{
        public :
                void print(){
                        cout << "output from sub class" << endl ;
                }
};


int main(){
        printIntro("runtime polymorphism", "06:40 09-dec-2023") ;
        Sub object ;
        object.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 :

```cpp
#include <iostream>
using namespace std ;
class Student{
        string name ;
        int age ;

        public :
           void Stu_info(){
                   cout << "enter name of student : " ;
                   getline(cin,name) ;
                   cout << "enter age of student : " ;
                   cin >> age ;
           }
           void Show_info(){
                   cout << "Name : " << name << endl ;
                   cout << "Age : " << age << endl ;
           }
} ;

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : " << time << endl ;
        cout << "_____*_____" << endl ;
}

int main(){
        printIntro("pointer to object", "06:48 09-dec-2023") ;
        Student jitendra ;
        Student *ptr = &jitendra ;
        ptr->Stu_info() ;
        ptr->Show_info() ;
        return 0 ;
}
```
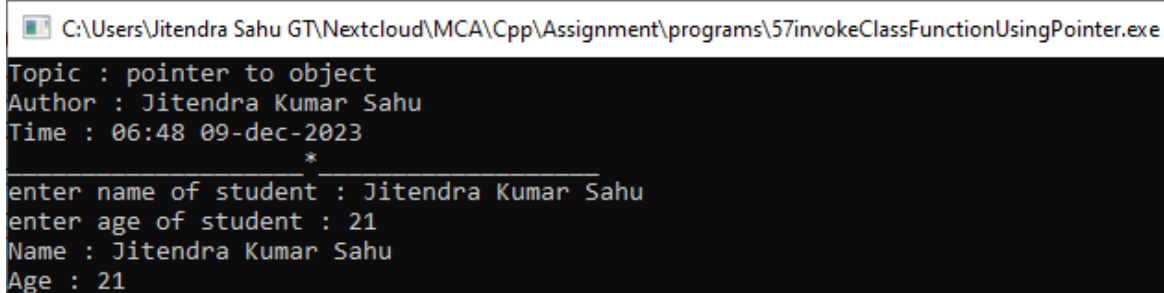
Output :

```
C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\57invokeClassFunctionUsingPointer.exe

Topic : pointer to object
Author : Jitendra Kumar Sahu
Time : 06:48 09-dec-2023
                  *_____
enter name of student : Jitendra Kumar Sahu
enter age of student : 21
Name : Jitendra Kumar Sahu
Age : 21
```
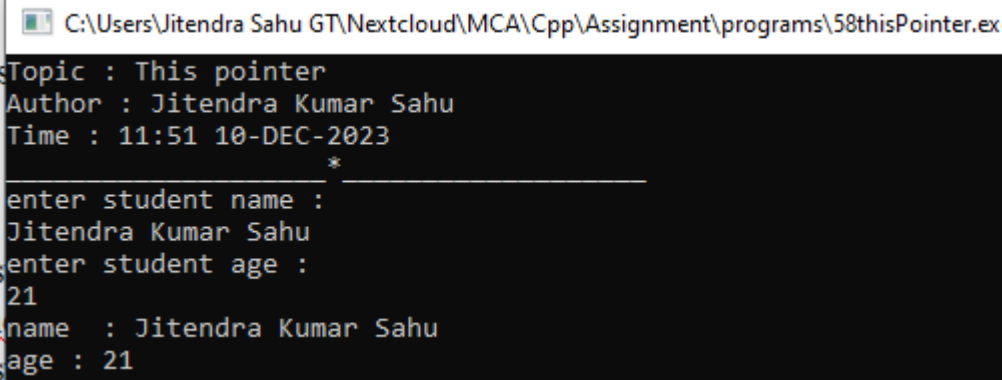
Program 58. Write a C++ program to illustrate functioning of this pointer.

Code :

```cpp
#include <iostream>
using namespace std ;

class Student{
        string name ;
        int age ;
        public :
                void setStdInfo(string name , int age) {
                        this->name = name ;
                        this->age = age ;
                }
                void printStdInfo(){
                        cout << "name  : " << name << endl ;
                        cout << "age : " << age << endl ;
                }
} ;

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : " << time << endl ;
        cout << "_____*_____" << endl ;
}

int main(){
        printIntro("This pointer", "11:51 10-DEC-2023") ;
        Student s1 ;
        string name ;
        int age ;
        cout << "enter student name : " << endl ;
        getline(cin,name)  ;
        cout << "enter student age : " << endl ;
        cin >> age ;
        s1.setStdInfo(name, age) ;
        s1.printStdInfo() ;
        return 0 ;
}
```

Output


```
Topic : This pointer
Author : Jitendra Kumar Sahu
Time : 11:51 10-DEC-2023
                    *
_____ _____
enter student name :
Jitendra Kumar Sahu
enter student age :
21
name  : Jitendra Kumar Sahu
age : 21
```

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 :

```cpp
#include <iostream>
using namespace std ;
class SecondClass ;
class FirstClass{
        public :
                int a ;
                FirstClass(){
                        this->a = 20 ;
                }
friend int multiply(FirstClass fobj, SecondClass sobj)  ;
} ;
class SecondClass{
        public :
                int b ;
                SecondClass(){
                        this->b = 20 ;
                }
friend int multiply(FirstClass fobj, SecondClass sobj)  ;
} ;

int multiply(FirstClass fobj, SecondClass sobj) {
         return fobj.a * sobj.b ;
}

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : " << time << endl ;
        cout << "_____*_____" << endl ;
}

int main(){
        printIntro("Friend function", "06:10 09-DEC-2023") ;
        FirstClass a ;
        SecondClass b ;
        cout <<"result of multiplication : " << multiply(a,b) << endl ;
        cout << "working! " << endl ;
        return 0 ;
}
```

Output :

C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\59friendFunction.exe

```
Topic : Friend function
Author : Jitendra Kumar Sahu
Time : 06:10 09-DEC-2023
_____*_____
result of multiplication : 400
working!
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

class Super{
    int var = 1 ;
        public :
            virtual void print(){
                cout << "value of var at super : " << var << endl ;
                }
};
class Sub : public Super {
    int var = 2 ;
        public :
            void print(){ // overridden function
                cout << "value of var at sub class : " << var << endl ;
                }
};

void printIntro(string topic, string time) {
cout<<"Topic : " << topic << endl ;
cout<<"Author : Jitendra Kumar Sahu" << endl ;
cout << "Time : "<< time << endl ;
cout <<"------------------*------------------"<< endl ;
}

int main(){

        printIntro("Pointer to derived class","05-12-23 18:05") ;
        Super *pointer ;  // Pointer of super class
        Sub object ;      // Sub class object

        pointer = &object ;
        pointer->print() ;
        return 0 ;

}
```
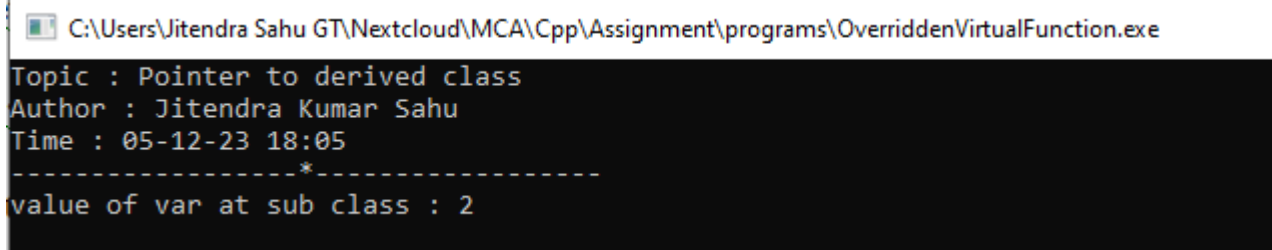
Output :


```
C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\OverriddenVirtualFunction.exe

Topic : Pointer to derived class
Author : Jitendra Kumar Sahu
Time : 05-12-23 18:05
------------------*------------------
value of var at sub class : 2
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

class Super{ // now the class whould be reffered as abstract class
    int var = 1 ;
        public :
        // pure virtual function
            virtual void print() = 0 ;
};
class Sub : public Super {
    int var = 2 ;
        public :
            void print(){ // overridding the function
                    cout << "value of var at sub class : " << var << endl ;
                    }
};

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"------------------*------------------"<< endl ;
}

int main(){

        printIntro("pure virtual function","05-12-23 18:05") ;
        Super *pointer ;  // Pointer of super class
        Sub object ;      // Sub class object

        pointer = &object ;
        pointer->print() ;
        return 0 ;

}
```
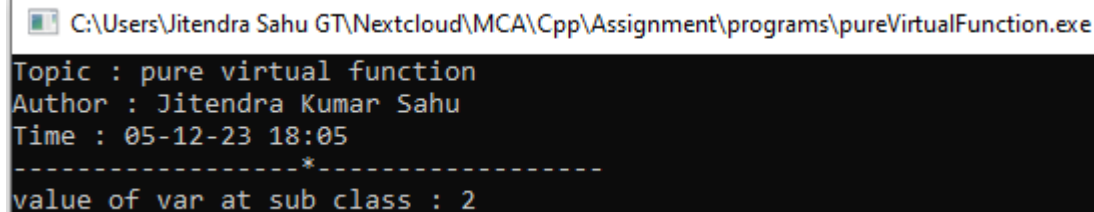
Output :


```
Topic : pure virtual function
Author : Jitendra Kumar Sahu
Time : 05-12-23 18:05
----------------*----------------
value of var at sub class : 2
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

class TwoNumbers{
 int a, b ;
        public :
         TwoNumbers(int a, int b) {
                this->a = a ;
                this->b = b ;
        }
        friend class Larger ;
} ;

class Larger{
        public :
        Larger(TwoNumbers tn){
                cout << "larger number is : " << endl ;
                tn.a > tn.b ? cout << tn.a : cout << tn.b ;
                cout << endl ;
        }
} ;

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl ;
        cout << "Author : Jitendra Kumar Sahu" << endl ;
        cout << "Time : " << time << endl ;
        cout << "_____*_____" << endl ;
}

int main(){
        printIntro("Program to demonstrait friend class","06:20 09-DEC-2023") ;
        int a , b ;
        cout << "enter two numbers : " ;
        cin >> a >> b ;
        TwoNumbers tw(a,b) ;
        Larger large(tw) ;
        return 0 ;
}
```
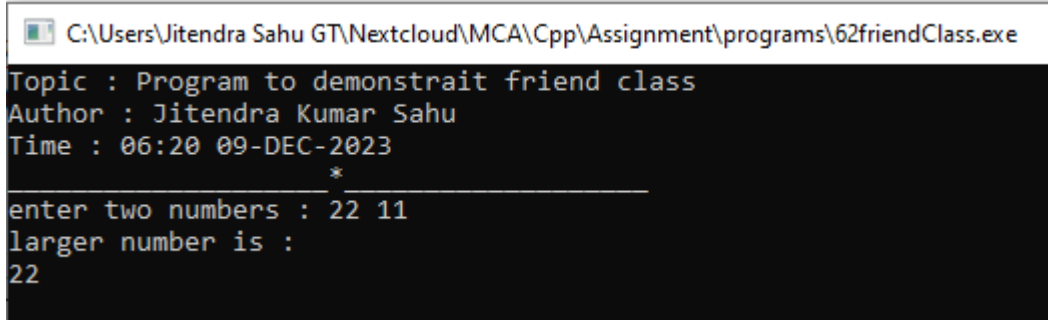
Output :

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

Code :

```cpp
#include <iostream>
using namespace std ;

class Complex{
        int a, b ;
        public :
        Complex(int a, int b){
                this->a = a ;
                this->b = b ;
        }
        bool operator == (const Complex c2){
                return (a == c2.a && b == c2.b) ;
        }
        void print(){
         cout << a << " + "<< b <<'i' << endl ;
        }
};

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"------------------*------------------"<< endl ;
}

int main(){
        printIntro("binary operator overloading","05-12-23 18:36") ;
        // creating three object of complex class
        Complex a(5,10) , b(10,20), c(10,20) ;

        cout << "a : " ;
        a.print();

        cout << "b : " ;
        b.print();

        cout << "c : " ;
        c.print();

        string res ;

        a == b ? res = "yes" : res = "no" ;
        cout << "a == b : " << res  << endl ;

        b == c ? res = "yes" : res = "no" ;
        cout << "b == c : " << res  << endl ;

        return 0 ;
}
```

Output :

C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\63binaryOperatorOverloadingIsEqual.exe

```
Topic : binary operator overloading
Author : Jitendra Kumar Sahu
Time : 05-12-23 18:36
------------------*------------------
a : 5 + 10i
b : 10 + 20i
c : 10 + 20i
a == b : no
b == c : yes
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

class Complex{
        int a, b ;
        public :
        Complex(int a, int b){
                this->a = a ;
                this->b = b ;
        }
        void operator ++ (){
                a++ ;
                b++ ;
        }
        void print() {
                cout << a << " + " << b << 'i' << endl ;
        }
};

void printIntro(string topic, string time) {
cout<<"Topic : " << topic << endl ;
cout<<"Author : Jitendra Kumar Sahu" << endl ;
cout << "Time : "<< time << endl ;
cout <<"-----------------*------------------"<< endl ;
}

int main(){

        printIntro("unary operator overloading","05-12-23 18:36") ;

        Complex a(5,10);
        cout << "before increament : " << endl ;
        a.print() ;
        ++a ;
        cout << "after increament : " << endl ;
        a.print() ;
        return 0 ;
}
```
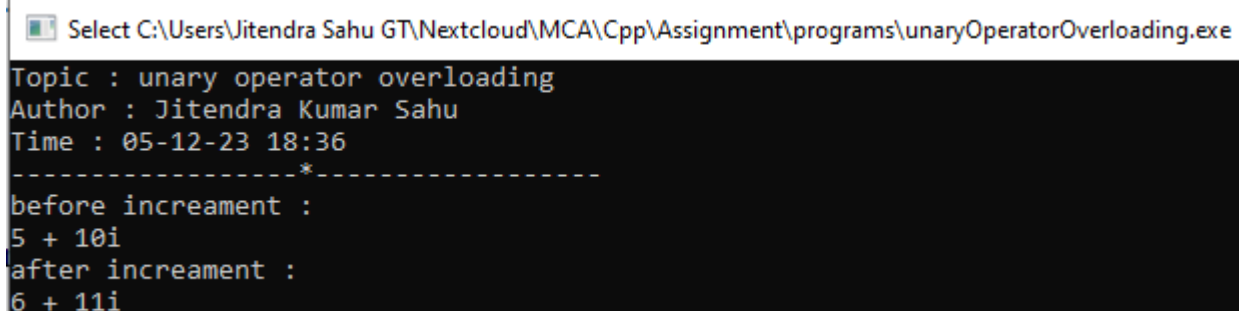
Output :

Jitendra Kumar Sahu   Page 82

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

Code :

```cpp
#include <iostream>
using namespace std ;

class Complex{
        int a, b ;
        public :
        Complex(){}
        Complex(int a, int b){
                this->a = a ;
                this->b = b ;
        }

friend Complex operator + (const Complex c1, const Complex c2);
        void print(){
         cout << a << " + "<< b <<'i' << endl ;
        }
};

Complex operator + (const Complex c1, const Complex c2){
                Complex c ;
                c.a = c1.a + c2.a ;
                c.b = c1.b + c2.b ;
                return c ;
}
void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"------------------*------------------"<< endl ;
}
int main(){

        printIntro("operator overloading using friend function","05-12-23 18:36") ;

        Complex a(5,10) , b(10,20);
        Complex c = a + b ;
        cout << "a : " ;
        a.print();

        cout << "b : " ;
        b.print();

        cout << "c i.e. a + b = " ;
        c.print();
        return 0 ;
}
```
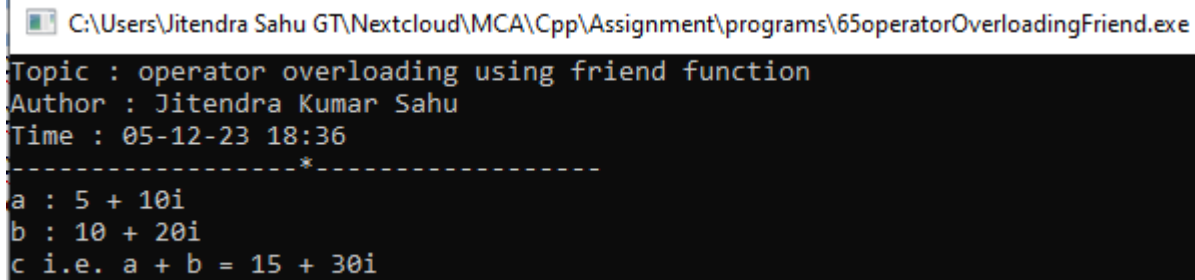
Output :

C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\65operatorOverloadingFriend.exe

```
Topic : operator overloading using friend function
Author : Jitendra Kumar Sahu
Time : 05-12-23 18:36
------------------*------------------
a : 5 + 10i
b : 10 + 20i
c i.e. a + b = 15 + 30i
```

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

Code :

```cpp
#include <iostream>
using namespace std ;

class TwoNumber{
        public :
        int a, b ;
        TwoNumber(){} // default constructor

        TwoNumber(int a, int b){
                this->a = a ;
                this->b = b ;
        }
// declration of friend operator function
friend TwoNumber operator - (const TwoNumber c1);
        void print(){
         cout <<"a : " <<  a << endl <<"b : "<< b << endl ;
        }
};

// definition of friend operator function
TwoNumber operator - (TwoNumber c1){
                c1.a =  -c1.a ;
                c1.b =  -c1.b ;
                return c1 ;
}

void printIntro(string topic, string time) {
 cout<<"Topic : " << topic << endl ;
 cout<<"Author : Jitendra Kumar Sahu" << endl ;
 cout << "Time : "<< time << endl ;
 cout <<"------------------*------------------"<< endl ;
}

int main(){
        printIntro("operator overloading using friend function","05-12-23 18:36") ;

        TwoNumber object(5,10);
        cout << "object : \n"  ;
        object.print();

        object = -object ;
        cout << "-object \n"  ;
        object.print();
        return 0 ;
}
```
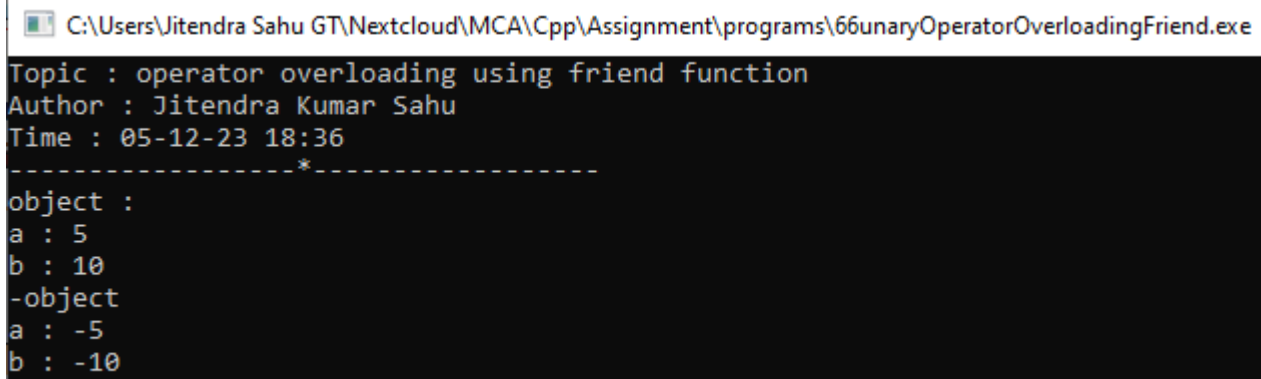
Output :

```
C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\66unaryOperatorOverloadingFriend.exe

Topic : operator overloading using friend function
Author : Jitendra Kumar Sahu
Time : 05-12-23 18:36
-----------------*-----------------
object :
a : 5
b : 10
-object
a : -5
b : -10
```

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

Code :

```cpp
#include <fstream>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

int main() {
        printIntro(
                  "program to illustrate\nextraction operator and getline() function.",
                  "13-12-23 21:36");
        try {
                ifstream myFile("Info.txt", ios::in);
                string s1, s2;

                getline(myFile, s1);  // use of getline()
                cout << "content read using getline from file : \n" << s1 << endl;

                cout << "content read using >> operator : \n"; // example of extraction
                                                // operator (>>)
                myFile >> s2;
                cout << s2 << endl;
        } catch (const std::exception& e) {
                cout << "Exception : ";
                std::cerr << e.what() << '\n';
  }

  return 0;
}
```
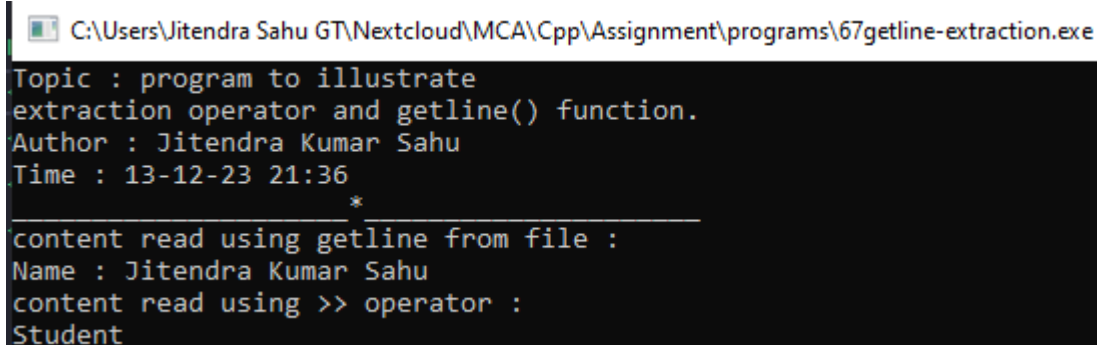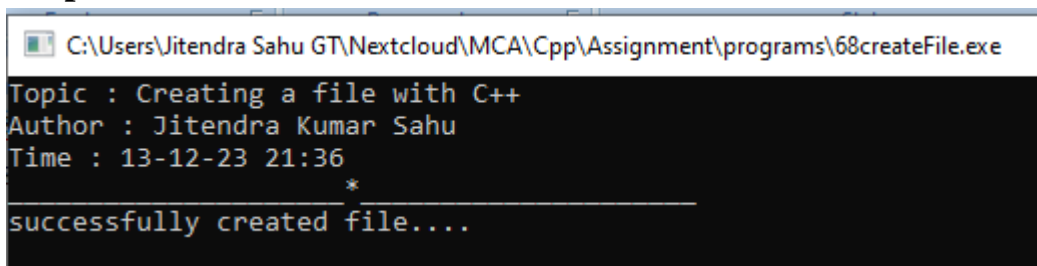
Output :

Program 68. WAP to create a file named "Rudra" using constructor.

Code :

```
#include <iostream>
#include <fstream>
using namespace std;

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

int main(){
        printIntro("Creating a file with C++", "13-12-23 21:36");
        try{
                // creating a file with constructor
                fstream myFile("Rudra.txt", ios::out);
                cout << "successfully created file...." << endl;
        }
        catch (const std::exception &e){
                cout << "Unable to create file." << endl;
                cout << "Exception : " << e.what() << endl;
        }

        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().
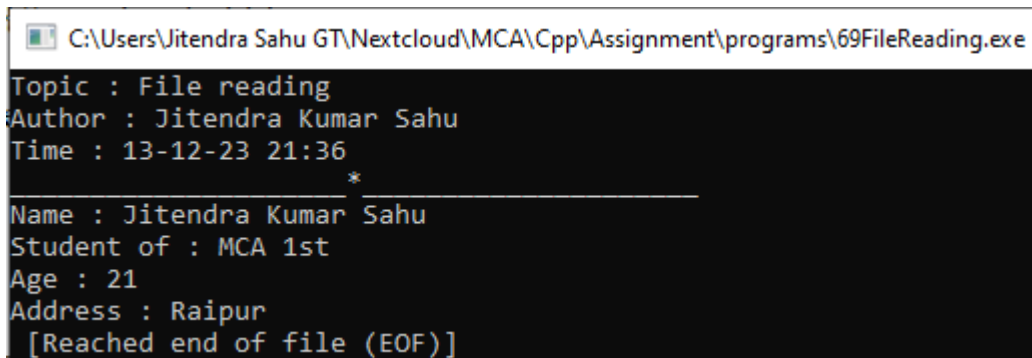
Code :

```cpp
#include <iostream>
#include <fstream>
using namespace std;

void printIntro(string topic, string time){
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

int main(){
        printIntro("File reading", "13-12-23 21:36");
        try{
                fstream myFile("Info.txt");
                char c;
                while (c = myFile.get()){
                    cout << c;
                    if (c == EOF) {//use of end of file

                        cout << "[Reached end of file (EOF)]";
                        break;
                    }
                }
        }
                catch (const std::exception &e){
                    cout << "Exception : " << e.what() << endl;
        }
    return 0;
}
```

Output :

Program 70.WAP to perform truncate operation in a file existing file named "File".
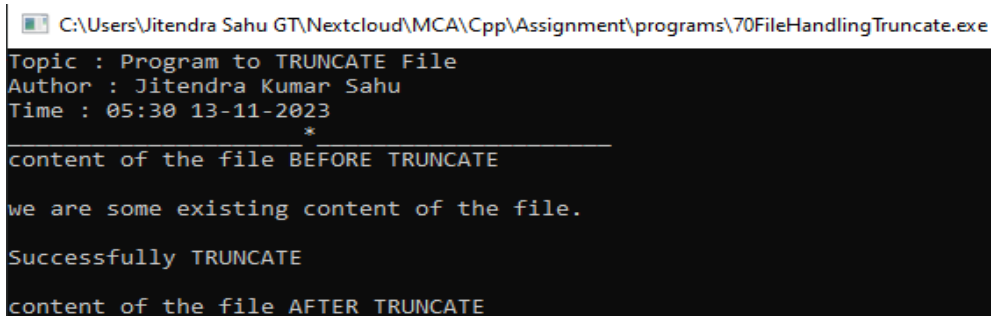
Code:

```cpp
#include <fstream>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

int main() {
        printIntro("Program to TRUNCATE File", "05:30 13-11-2023");
        const char fileName[5] = "File" ;
        try {
                ifstream myFile(fileName, ios::in);  // opening file in reading mode
                 cout << "content of the file BEFORE TRUNCATE\n" << endl;
                 string s;
                 while (getline(myFile, s)) {  // reading content line by line
                     cout << s;                // printing content of line
                 }
                 cout << endl << endl;
                 myFile.close();

                 ofstream file(fileName, ios::trunc);  // open file with truncate
                 cout << "Successfully TRUNCATE\n" << endl;
                 myFile.close();

                 cout << "content of the file AFTER TRUNCATE\n" << endl;
                 while (getline(myFile, s)) {
                         cout << s;
                 }
                 cout << endl;
                 myFile.close();
        } catch (const exception &e) {
                 cerr << "Sorry something went wrong! \n";
                 cerr << e.what() << endl;
        }
        return 0;
}
```

Output :

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

Code:

```cpp
#include <fstream>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}
int main() {
        const char fileName[10] = "Shiva.txt";
        printIntro("Program to append content File", "05:30 13-11-2023");
        try {
                ifstream myFile(fileName, ios::in);  // opening file in reading mode
                cout << "content of the file BEFORE APPEND\n" << endl;
                string s;
                while (getline(myFile, s)) {  // reading content line by line
                        cout << s;              // printing content of line
                }
                cout << endl << endl;
                myFile.close();
        } catch (const exception &e) {
                cerr << "Sorry something went wrong! \n";
                cerr << e.what() << endl;
        }
        try {
                ofstream myFile(fileName, ios::app);  // open file with truncate
                // APPENDING content to file
                myFile << "\nWE ARE THE NEW CONTENT APPENDED TO FILE SHIVA.TXT\n";
                cout << "Successfully APPENDED!\n" << endl;
                myFile.close();
        } catch (const std::exception &e) {
                std::cerr << e.what() << '\n';
        }
        try {
                ifstream myFile(fileName, ios::in);
                cout << "content of the file AFTER Append\n" << endl;
                string s;
                while (getline(myFile, s)) {
                        cout << s;
                }
                cout << endl << endl;
                myFile.close();
        } catch (const std::exception &e) {
                std::cerr << e.what() << '\n';
        }

        return 0;
}
```

Output :

Select C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\71fileAppendContent.exe

```
Topic : Program to append content File
Author : Jitendra Kumar Sahu
Time : 05:30 13-11-2023
_____*_____
content of the file BEFORE APPEND

we are the content inside file Shiva.txt.

Successfully APPENDED!

content of the file AFTER Append

we are the content inside file Shiva.txt.WE ARE THE NEW CONTENT APPENDED TO FILE SHIVA.TXT
```

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

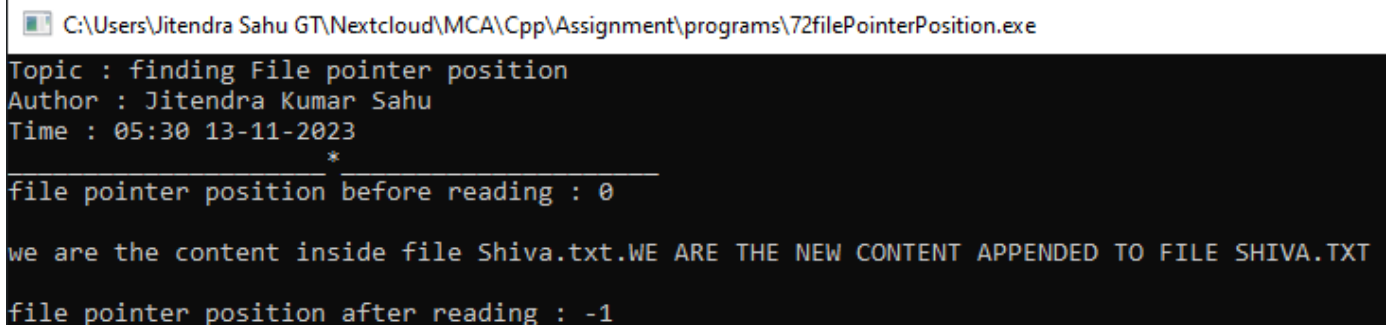**Code:**

```cpp
#include <fstream>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

int main() {
        const char fileName[10] = "Shiva.txt";
        printIntro("finding File pointer position", "05:30 13-11-2023");
        try {
                ifstream myFile(fileName, ios::in);  // opening file in reading mode
                // printing file pointer postion
                string s;
                cout << "file pointer position before reading : " << myFile.tellg() << endl << endl;
                while (getline(myFile, s)) {  // reading content line by line
                  cout << s;               // printing content of line
                }
                // printing file pointer postion
                cout << "\n\nfile pointer position after reading : " << myFile.tellg() << endl;
                cout << endl ;
                myFile.close();
        } catch (const exception& e) {
                cerr << "Sorry something went wrong! \n";
                cerr << e.what() << endl;
        }

        return 0;
}
```

Output :


```
C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\72filePointerPosition.exe

Topic : finding File pointer position
Author : Jitendra Kumar Sahu
Time : 05:30 13-11-2023
                    *
file pointer position before reading : 0

we are the content inside file Shiva.txt.WE ARE THE NEW CONTENT APPENDED TO FILE SHIVA.TXT

file pointer position after reading : -1
```

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

Code:

```cpp
#include <fstream>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "_____*_____" << endl;
}

int main() {
        printIntro("Program to differentiate read() and getLine()", "05:30 13-11-2023");
        const char fileName[10] = "Shiva.txt";
        char buff[11];
        buff[10] = '\0';
        try {
                ifstream myFile(fileName, ios::in);  // opening file in reading mode
                myFile.read(buff, 10);              // reading 10 characters using read()
                cout << "content of the file using read() : " << endl;
                cout << buff << endl;

                cout << "content of the file using getline() : " << endl;
                string s;
                while (getline(myFile, s)) {  // reading content line by line
                    cout << s;                // printing content of line
                }
                cout << endl << endl;
                myFile.close();
        } catch (const exception& e) {
                cerr << "Sorry something went wrong! \n";
                cerr << e.what() << endl;
        }

        return 0;
}
```
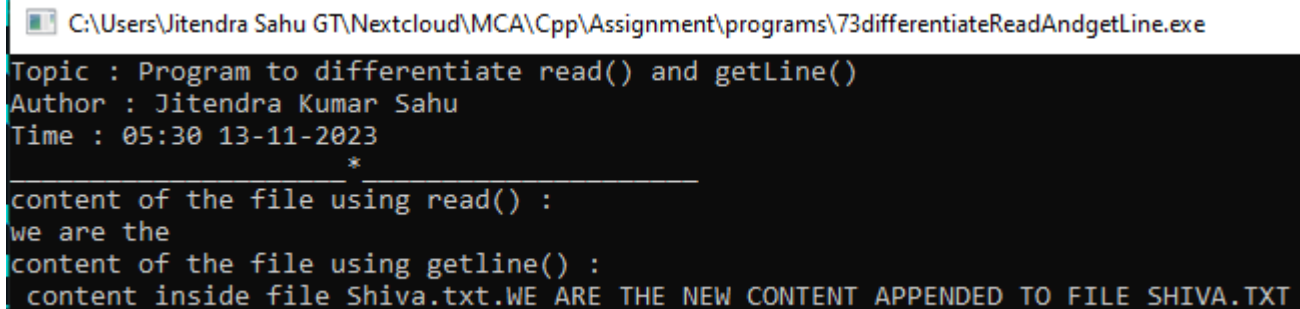
Output :

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

Code:

```cpp
#include <iomanip>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "-------------------*--------------------" << endl;
}

int main() {
        printIntro("Program to demonstrate some iomanip functions",
                "05:30 13-11-2023");
        const int num = 123;
        const double pi = 3.14159265358979323846;

        cout << "setw() \n";
        cout << setw(10) << num << endl;
        cout << setw(10) << pi << endl;

        cout << "\nsetprecision() \n";
        cout << setprecision(6) << num << endl;
        cout << setprecision(6) << pi << endl;

        cout << "\nsetbase() \n";
        cout << setbase(16) << num << endl;
        cout << setbase(8) << pi << endl;

        cout << "\nsetfill() \n";
        cout << setfill('*')<< setw(10) << num << endl;
        cout << setfill('#')<< setw(10) << pi << endl;

        return 0;
}
```
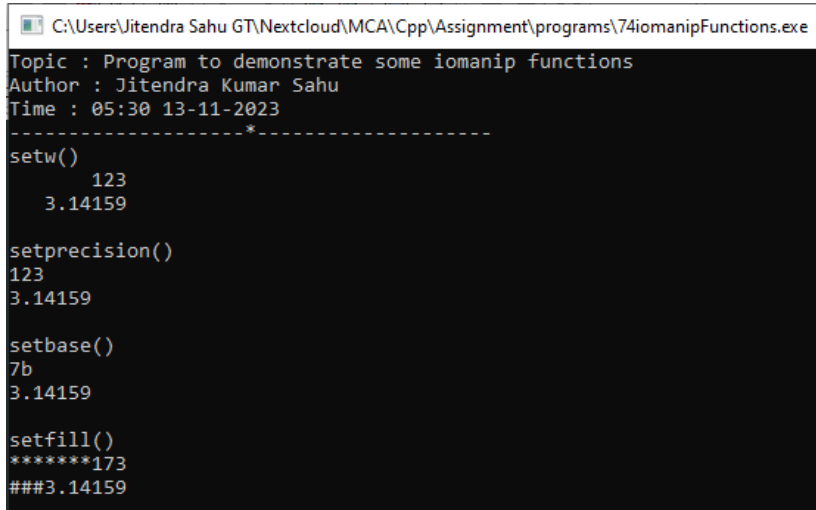
Output :


```
C:\Users\Jitendra Sahu GT\Nextcloud\MCA\Cpp\Assignment\programs\74iomanipFunctions.exe

Topic : Program to demonstrate some iomanip functions
Author : Jitendra Kumar Sahu
Time : 05:30 13-11-2023
-------------------*-------------------
setw()
       123
   3.14159

setprecision()
123
3.14159

setbase()
7b
3.14159

setfill()
*******173
###3.14159
```

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.
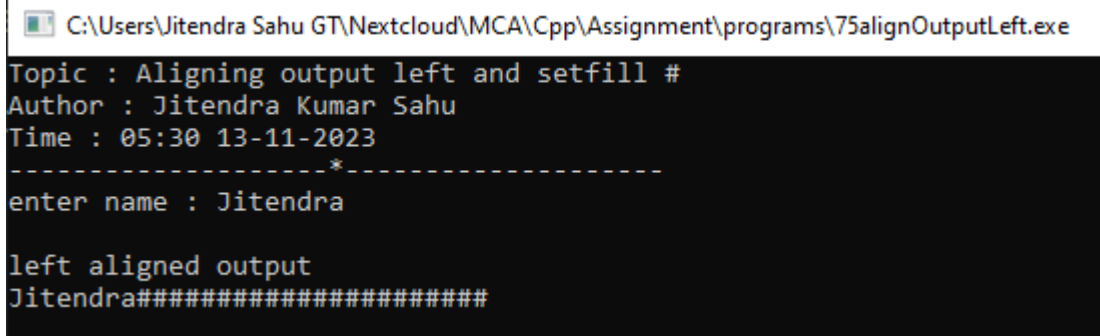
Code:

```cpp
#include <iomanip>
#include <iostream>
using namespace std;

void printIntro(string topic, string time) {
        cout << "Topic : " << topic << endl;
        cout << "Author : Jitendra Kumar Sahu" << endl;
        cout << "Time : " << time << endl;
        cout << "--------------------*--------------------" << endl;
}

int main() {
        printIntro("Aligning output left and setfill #",
                "05:30 13-11-2023");
        string name ;
        cout << "enter name : " ;
        getline(cin,name) ;
        cout << "\nleft aligned output\n";

        cout << left << setw(30) << setfill('#')<< name << endl;
        return 0;
}
```

Output :