Revision
And
Sample Questions for Practice

Arithmetic Operators:

Here is a list of Arithmetic Operators that can be used.

Operator	Meaning	Example
+	Addition	8+2=10
-	Subtraction	8-2=6
*	Multiplication	8*2=16
/	Division	8/2=4
%	Modulus	8%2=0

Expression : Precedence Order

Here is the precedence order of Arithmetic Operators

Operator	Symbol	Precedence
Parentheses	( )	1
Exponential	X^y	2
Multiplication Division	* /	3 3
Addition Subtraction	+ -	4 4

Working Example : Precedence Order

What will be the Output?

100 / 10 * 10

1

or

100

Associativity of Operators

Operators Associativity is used when two operators of same precedence appear in an expression. Associativity can be either Left to Right or Right to Left.

Symbol	Operator	Associativity
* / %	Multiplication/division/modulus	left-to-right
+ -	Addition/subtraction	left-to-right
=	Assignment	right-to-left

Working Example : Precedence Order

What will be the Output?

100 + 200 / 10 - 3 * 10 % 10

Logical Operators: Precedence Order

Precedence Order	Operator	In C++
1	Not	!
2	AND	&&
3	OR	||

Working Example :

What will be the Output?
If a = 1; b = 6; and c = 3;

```
a || (b * c);
a && (b < c);
```

Working Example :

What will be the Output?

```
int a = 5;
int b = 9;
```

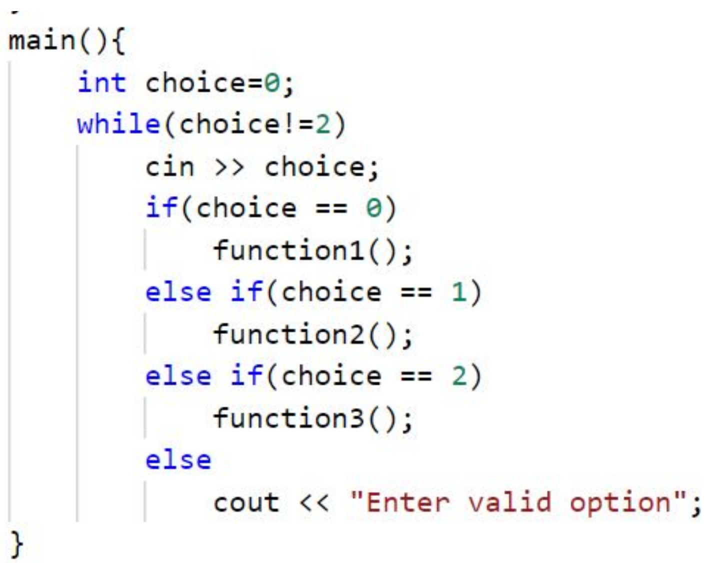
```
cout << ((a == 0) && (a > b)) << endl;
cout << ((a == 0) && (a < b)) << endl;
cout << ((a == 0) || (a > b)) << endl;
cout << ((a == 0) || (a < b)) << endl;
cout << !(a == 0) << endl;
cout << !(a == 5) << endl;
```

Working Example: Functions

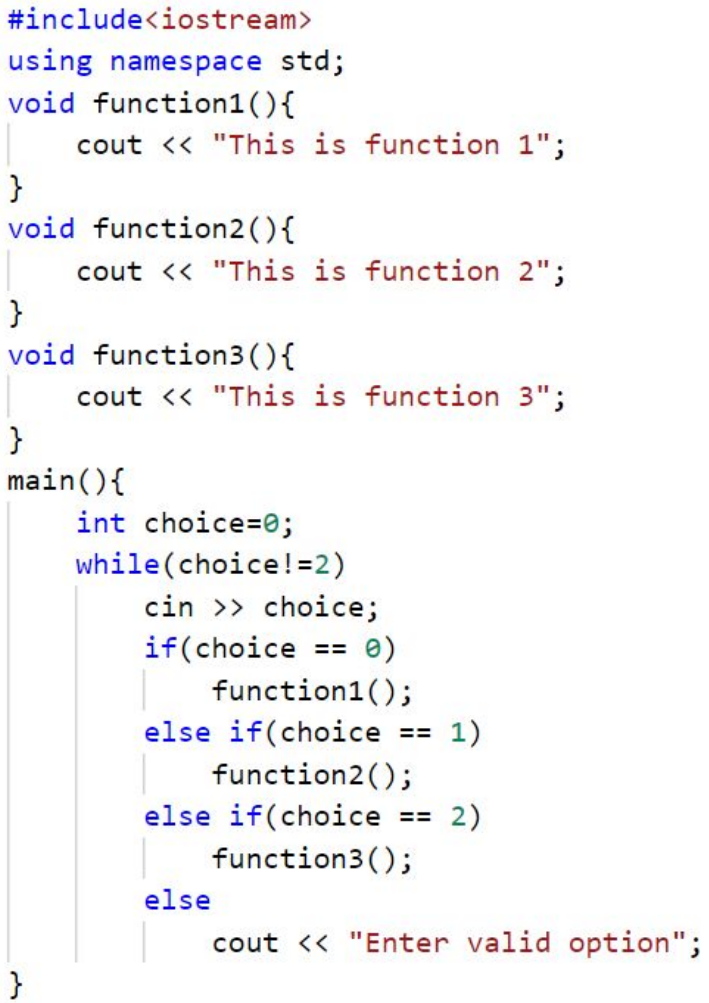
What will be the Output?

```
int x = 0;
int f1(){
      x = 5;
      return x;
}
int f2(){
      x = 10;
      return x;
}
main(){
      int p = f1() + f2();
      cout << p << x;
}
```

Working Example



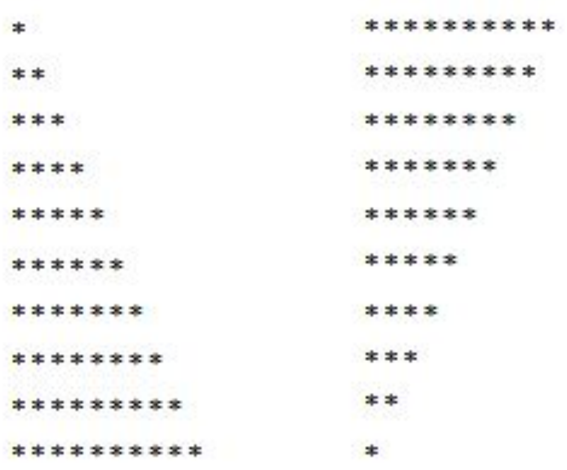
```
main(){
    int choice=0;
    while(choice!=2)
        cin >> choice;
        if(choice == 0)
            function1();
        else if(choice == 1)
            function2();
        else if(choice == 2)
            function3();
        else
            cout << "Enter valid option";
}
```



```
#include<iostream>
using namespace std;
void function1(){
    cout << "This is function 1";
}
void function2(){
    cout << "This is function 2";
}
void function3(){
    cout << "This is function 3";
}
main(){
    int choice=0;
    while(choice!=2)
        cin >> choice;
        if(choice == 0)
            function1();
        else if(choice == 1)
            function2();
        else if(choice == 2)
            function3();
        else
            cout << "Enter valid option";
}
```

Working Example

Write a C++ program separately that prints the following patterns separately one below the other. Use nested for loops to generate the patterns.



```
*                   **********
**                  *********
***                 ********
****                *******
*****               ******
******              *****
*******             ****
********            ***
*********           **
**********          *
```

Working Example

A number is said to be Harshad if it's exactly divisible by the sum of its digits. Create a function that determines whether a number is a Harshad or not.

- isHarshad(75) ➞ false
  // 7 + 5 = 12
  // 75 is not exactly divisible by 12

- isHarshad(171) ➞ true
  // 1 + 7 + 1 = 9
  // 9 exactly divides 171

- isHarshad(481) ➞ true

Working Example

The iterated square root of a number is the number of times the square root function must be applied to bring the number strictly under 2.
Given an integer, return its iterated square root. Return -1 if it is negative.

- iSqrt(1) → 0
- iSqrt(2) → 1
- iSqrt(7) → 2
- iSqrt(27) → 3
- iSqrt(256) → 4
- iSqrt(-256) → -1

Working Example

Write a function that takes a number from the user and return whether the number is prime number or not.

Prime number are those who only divisible by 1 and its own.

Some example of prime numbers are 2, 3, 5, 7, 11, 13, 17 etc.

```
bool isPrime(int number); // function header
```