CS101: Problem Solving through C Programming

Sachchida Nand Chaurasia

Assistant Professor

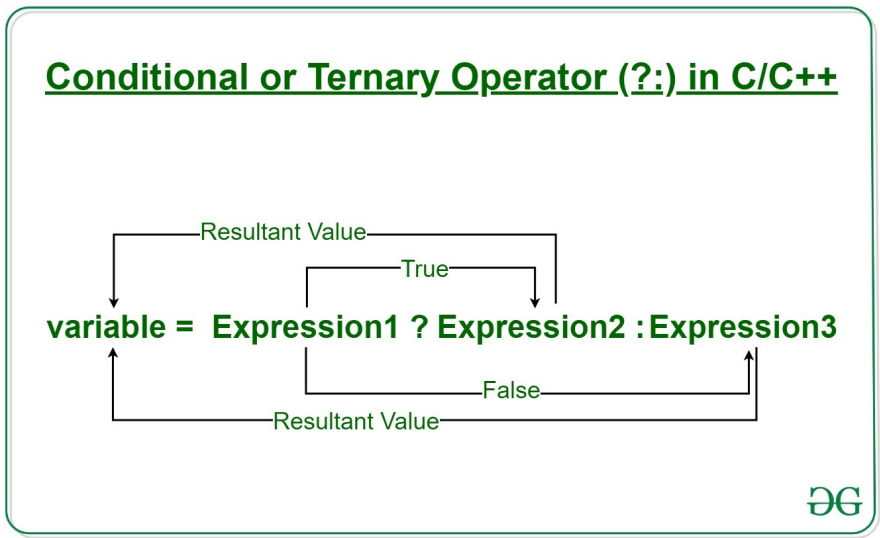
Department of Computer Science
Banaras Hindu University
Varanasi
Email id: snchaurasia@bhu.ac.in, sachchidanand.mca07@gmail.com

February 11, 2021

Ternary Operator I

Conditional or Ternary Operator (?:)

✓ The conditional operator is kind of similar to the if-else statement as it does follow the same algorithm as of if-else statement but the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.



Meaning of the above syntax.

Ternary Operator II

- In the above syntax, the expression1 is a Boolean condition that can be either true or false value.
- If the expression1 results into a true value, then the expression2 will execute.
- The expression2 is said to be true only when it returns a non-zero value.
- If the expression1 returns false value then the expression3 will execute.
- The expression3 is said to be false only when it returns zero value.

Syntax: The conditional operator is of the form

Ternary Operator III

```
variable = Expression1 ? Expression2 : Expression3
It can be visualized into if-else statement as:

if(Expression1)
{
      variable = Expression2;
}
else
{
      variable = Expression3;
}
```

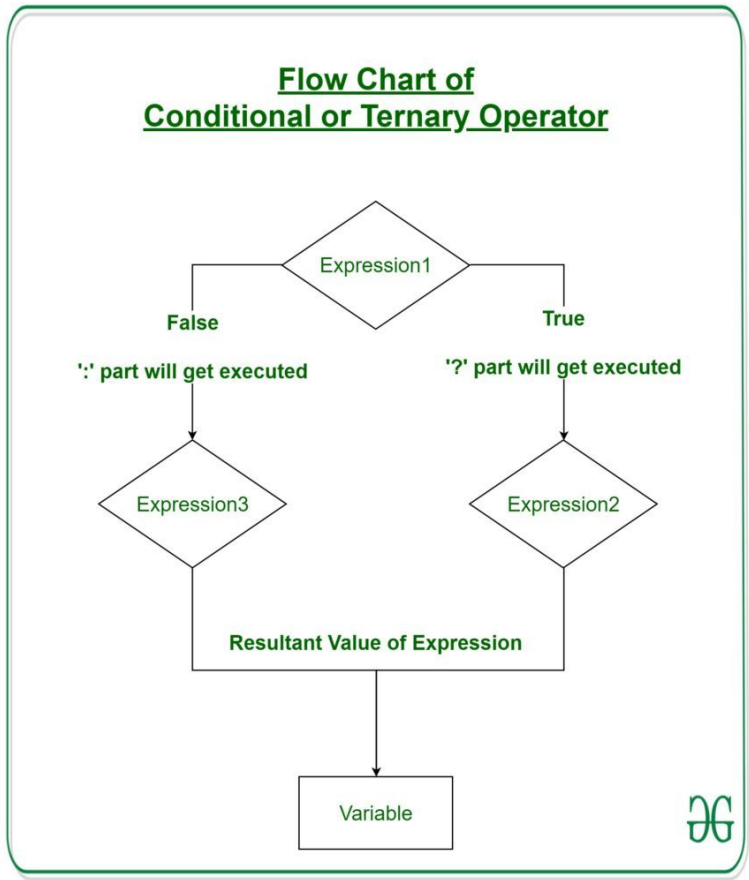
✓ Since the Conditional Operator '?:' takes three operands to work, hence they are also called ternary operators.

Working:

Here, Expression1 is the condition to be evaluated. If the condition(Expression1) is True then Expression2 will be executed and

Ternary Operator IV

the result will be returned. Otherwise, if the condition(Expression1) is false then Expression3 will be executed and the result will be returned.



Ternary Operator V

```c
// C program to find largest among two
// numbers using ternary operator
#include <stdio.h>
int main()
{
	int m = 5, n = 4;
	(m > n) ? printf("m is greater than n that is %d > %d", m, n): printf("n is greater than m that is %d
	 > %d",n, m);
	return 0;
}
```

```c
#include <stdio.h>
int main()
{
	int age;  // variable declaration
	printf("Enter your age");
	scanf("%d",&age);   // taking user input for age variable
	(age>=18)? (printf("eligible for voting")) : (printf("not eligible for voting"));  // conditional
	operator
	return 0;
}
```

Ternary Operator VI

```c
#include <stdio.h>
int main()
{
          int a=5,b;  // variable declaration
          b=((a==5)?(3):(2)); // conditional operator
          printf("The value of 'b' variable is : %d",b);
       return 0;
}
```

- A conditional operator is a single programming statement, while the 'if-else' statement is a programming block in which statements come under the parenthesis.
- A conditional operator can also be used for assigning a value to the variable, whereas the 'if-else' statement cannot be used for the assignment purpose.

Ternary Operator VII

- It is not useful for executing the statements when the statements are multiple, whereas the 'if-else' statement proves more suitable when executing multiple statements.
- The nested ternary operator is more complex and cannot be easily debugged, while the nested 'if-else' statement is easy to read and maintain.

```
#include<stdio.h>
#include<conio.h>

void main()
{
      int a, b, c, large;
      clrscr();
      printf("Enter any three number: ");
      scanf("%d%d%d",&a,&b,&c);
      large=a>b ? (a>c?a:c) : (b>c?b:c);
      printf("Largest Number is: %d",large);
      getch();
}
```

Ternary Operator VIII

```c
#include<stdio.h>
int main()
{
	int a=2,b=4,c=9;
	int lar;
	lar=(((a>b)&&(a>c))?a:((b>c)?b:c));
	printf("Largest Number is: %d",lar);
	return 0;
}
```

```c
#include<stdio.h>
int main()
{
	int num;
	scanf("%d",&num);
	num < 0 ? printf("Negative") : printf("Positive");
	return 0;
}
```

Ternary Operator IX

```c
#include <stdio.h>
main()
{
       int a , b;
       a = 10;
       printf( "Value of b is %d\n", (a == 1) ? 20: 30 );
       printf( "Value of b is %d\n", (a == 10) ? 20: 30 );
}
```