Tutorial Link https://codequotient.com/tutorials/Conditional Operator/5a22a02bc66cfe38f2962242

TUTORIAL

Conditional Operator

Chapter

1. Conditional Operator

This is an operator defined as an quick way for short if-else constructs. It has following form: -

```
expr1 ? expr2 : expr3;
```

All are expression, which will be executed. In this, expr1 will be evaluated as a condition, if it evaluates to true, expr2 will be executed, otherwise expr3 will be executed. The above code is another form of following if-else block: -

```
if(expr1)
   expr2;
else
   expr3;
```

So, conditional operator is a kind of compact form to represent some if-else constructs. For example,

```c
#include <stdio.h>

int main()
{
  int marks = 49;
  char grade;
  if(marks >= 40)
  {
    grade='P';
  }
  else
  {
    grade='F';
  }
  printf("You got %c grade.",grade);
  return 0;
}

```

The above program can be written as:

```c
#include <stdio.h>

int main()
{
  int marks = 49;
  char grade;

  grade = (marks >= 40) ? 'P' : 'F';    // if-else as conditional operator.

  printf("You got %c grade.",grade);
  return 0;
}

```

It is quite useful sometimes to write with conditional operator instead the if-else block as it makes the code compact and easy to read. We can make the things bit complicated to write nested if or nested if-else using conditional operators. One example to illustrate the use is:

```c
#include <stdio.h>

int main()
{
  int marks = 57;
  char grade;

  if(marks >= 40)
  {
    if(marks < 60)
    {
      grade = 'B';
    }
    else
    {
      grade = 'A';
    }
  }
  else
  {
    grade = 'F';
  }
  printf("You got %c grade.",grade);
  return 0;
}

```

The above program can be written with nested conditional operators as:

```c
#include <stdio.h>

int main()
{
  int marks = 7;
  char grade;

  grade = (marks >= 40) ? (marks < 60) ? 'B' : 'A' : 'F';

  printf("You got %c grade.",grade);
  return 0;
}
```