Lab 5-2 : Conditional Statements in C Language contd. : Ternary Operators & ‘Switch’ Statement

*Familiarization with conditional operators, and decisions using the switch statement.*

# The Conditional Operators

The conditional operators **?** and : are sometimes called ‘ternary operators’ because they take three arguments. The first argument is a comparison argument, the second is the result upon a true comparison, and the third is the result upon a false comparison. If it helps you can think of the operator as shortened way of writing an if-else statement. It is often used as a way to assign variables based on the result of a comparison. When used correctly it can help increase the readability and reduce the amount of lines in your code.

## Syntax:

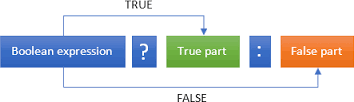


Figure 1 - The syntax of ternary operators

What this expression says is: “if expression 1 is true (that is, if its value is non-zero), then the value returned will be True part, otherwise the value returned will be False part”. Let us understand this with the help of a few examples:

## EXAMPLE 1

1

#include<stdio.h>

**int** main()

{

**int** num;

printf("Enter the Number : "); scanf("%d",&num); (num%2==0)?printf("Even"):printf("Odd");

}

Programming Fundamentals

## EXAMPLE 2

Let’s say that we apply a higher shipping charge of PKR 1500.00 to orders placed in Karachi. If our order is not placed in Karachi, then the shipping charge is PKR 1000.00. The following example shows the if-else statement and the equivalent ternary statement.

**USING “IF ELSE”**

//if-else

if(city = = 'K')

{

shipping\_charge=1500.00;

}

else

{

shipping\_charge=1000.00;

}

**USING “TERNARY OPERATOR”**

//ternary operator

shipping\_charge = (city = = 'K') ? 1500.00 : 1000.00;

As you can see, what before was 8 lines of code has been reduced to one line.

The ternary operator can also be nested so that different conditions can be tested based on a result of true.

# Decisions using the SWITCH statement

The control statement that allows us to make a decision from the number of choices is called switch, or more correctly a switch-case-default, since these three keywords go together to make up the control statement. This means that it is used to select one of the several alternatives such as menu options. The switch statement is especially useful when the selection is based on the value of a single variable called the *controlling variable* or of a simple expression called the *controlling expression.*

## Syntax:

The general form of switch statement is:

Programming Fundamentals

2

**switch** ( integer expression )

{

**case** constant 1 : body1

**break**;

**case** constant 2 : body2 **break**;

**case** constant 3 : body3 **break**;

**default** :

**default**-body

}

next-statement;

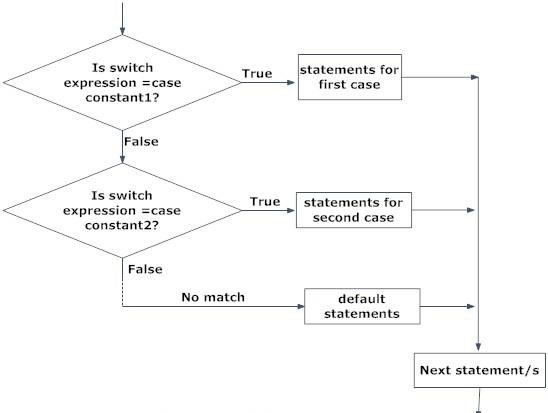


Figure 2 - The switch structure

The integer expression following the keyword switch is any C expression that will yield an integer value. It could be an integer constant like 1, 2 or 3, or an expression that evaluates to an integer. The keyword case is followed by an integer or a character constant. Each constant in each case must be different from all the others i.e. unique. The “body” lines in the above form of switch represent any valid C statement. What happens when we run a program containing a switch? First, the integer expression following the keyword switch is evaluated. The value it gives is then matched, one by one, against the constant values that follow the case statements. When a match is found, the program executes the statements following that case, and all subsequent case and default statements as well. If no match is found with any of the case statements, only the statements following the default are executed.

A few examples will show how this control structure works.

## EXAMPLE 3

Consider the following program:

**main( )**

**{**

**int i = 2 ;**

**switch ( i )**

**{**

**case 1 :**

**printf ( "I am in case 1 \n" ) ;**

**case 2 :**

**printf ( "I am in case 2 \n" ) ;**

**case 3 :**

**printf ( "I am in case 3 \n" ) ;**

**default :printf ( "I am in default \n" ) ;**

**}**

**}**

The output of this program would be:

**I am in case 2 I am in case 3 I am in default**

The output is definitely not what we expected! We didn’t expect the second and third line in the above output. The program prints case 2 and 3 and the default case. Well, yes. We said the switch executes the case where a match is found and all the subsequent cases and the default as well.

If you want that only case 2 should get executed, it is up to you to get out of the switch then and there by using a **break** statement. The following example shows how this is done. Note that there is no need for a break statement after the default, since the control comes out of the switch anyway.

## EXAMPLE 4

**main( )**

**{**

**int i = 2 ; switch ( i )**

**{**

**case 1 :**

**printf ( "I am in case 1 \n" ) ;**

**break ;**

**case 2 :**

**printf ( "I am in case 2 \n" ) ;**

**break ;**

**case 3 :**

**printf ( "I am in case 3 \n" ) ;**

**break ;**

**default :printf ( "I am in default \n" ) ;**

**}**

**}**

The output of this program would be:

**I am in case 2**

## EXAMPLE 5

**#include<stdio.h> main()**

**{**

**char myinput;**

**printf("Which option will you choose:\n"); printf("a) Program 1 \n");**

**printf("b) Program 2 \n");**

**scanf("%c", &myinput);**

**switch (myinput)**

**{**

**case 'a':**

**printf("Run program 1\n"); break;**

**case 'b':**

**{**

**printf("Run program 2\n"); break;**

**}**

**default:**

**printf("Invalid choice\n"); break;**

**}**

**}**

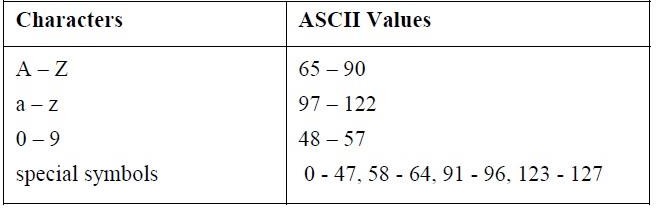
Student Name: Roll No: Section:

**Student Exercises and Tasks:**

## Task 1:

If the ages of Ali, Sarah and Ahmed are input through the keyboard, write a program to determine the youngest of the three, *using ternary operator.*

## Task 2:

Any character is entered through the keyboard; write a program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol.

## Task 3:

DUET has following grading scheme, write programming code to implement the grading scheme. Marks will be entered by the user:

|  |  |  |
| --- | --- | --- |
| **Grades** | **Marks** | **CGPA** |
| **A** | 86 | 4 |
| **A-** | 82 | 3.67 |
| **B+** | 78 | 3.33 |
| **B** | 74 | 3 |
| **B -** | 70 | 2.67 |
| **C+** | 66 | 2.33 |
| **C** | 62 | 2.00 |
| **C-** | 58 | 1.67 |
| **D+** | 54 | 1.33 |
| **D** | 50 | 1.00 |

## Task 4:

Write a program for quiz, where the system asks question and user will select answer from the given four options. On correct answer the program will congratulate and on wrong answer print sorry message.

Student Name: Roll No:

## Task 5:

Section:

A university has the following rules for a student to qualify for a degree with A as the main subject and B as the subsidiary subject:

* + - 1. He should get 55 percent or more in A and 45 percent or more in B.
      2. If he gets than 55 percent in A he should get 55 percent or more in B. However, he should get at least 45 percent in A.
      3. If he gets less than 45 percent in B and 65 percent or more in A he is allowed to reappear in an examination in B to qualify.
      4. In all other cases he is declared to have failed.

Write a program to receive marks in A and B and Output whether the student has passed, failed or is allowed to reappear in B. *(Using Ternary operator)*