Engr 101 **August 28, 2019**
Day 4

Selection Structures IF and SWITCH

In the previous chapters we have written programs which execute statements linearly one after another. This is called and *In-Line Structure* or sometime it is referred to as the *Compound Structure.*

A *Compound Structure* is a group of statements bracketed by { and } that are executed sequentially.

In this chapter we look at the *Selection Structure* which allows program to become two dimensional – that is, the no longer execute all of the code sequentially as it appears but may branch from one section of the code to another depending on calculated values within the code.

The selection structure depends on the value of a *Condition* which evaluates to true (1) or false (0). For example (a > b) is true or false.

The greater than (>) operator is called a *relational operator* in that it describes a relationship between two variable or expressions. There are other relational operators.

Relational and Equality Operators

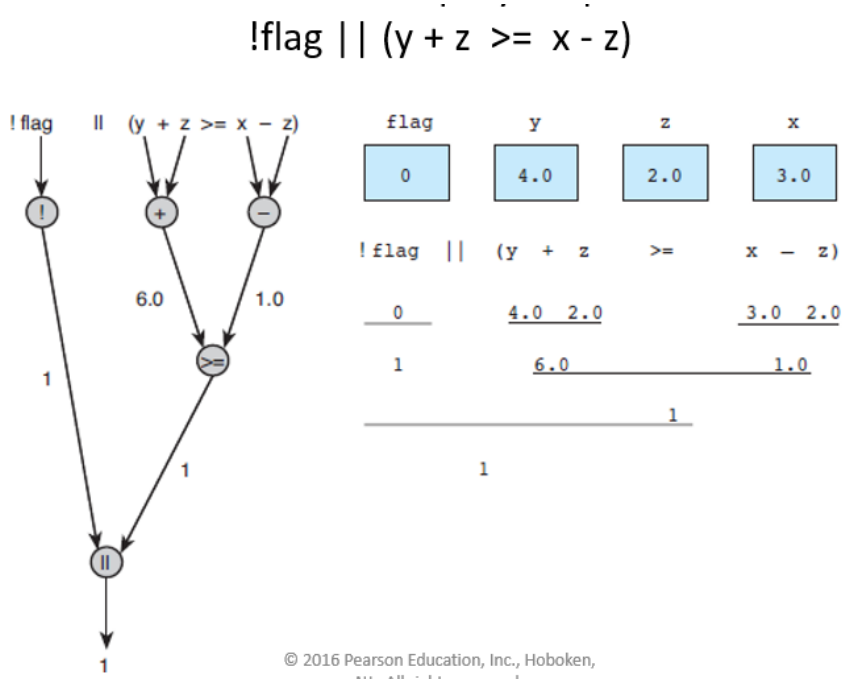
Operator	Meaning	Type
<	less than	relational
>	greater than	relational
<=	less than or equal to	relational
>=	greater than or equal to	relational
==	equal to	relational
!=	not equal to	equality

In addition to the relational operators there are also some logical operators

- logical expressions
  - an expression that uses one or more of the logical operators
    - && (and)
    - || (or)
    - ! (not)

Do examples with AND, OR, and NOT

Operator	Precedence
function calls	highest (evaluated first)
! + - & (unary operator)	
* / %	
+ -	
< <= >= >	
== !=	
&&	
||	
=	lowest (evaluated last)



Note that a logical expression stops being evaluated as soon as the outcome can be determined.

For example, suppose x = 0
(x != 0 && y/x > 5) will be OK but
(y/x > t && x != 0) will fail

Also, we can compare characters with logical expressions:

Expression	Value
'9' >= '0'	1 (true)
'a' < 'e'	1 (true)
'B' <= 'A'	0 (false)
'Z' == 'z'	0 (false)
'a' <= 'A'	System dependent
'a' <= ch && ch <= 'z'	1 (true) if ch is a lowercase letter

The IF statement is a selection structure that makes use of a logical expression. Its general form is:

```
if(logical expression)
  {
    statements to do if
    logical expression is true
  }
else
  {
   statements to do if
    logical expression is false.
  }
```

There are some variations and options on this general form:

If statement with one alternative

```
if  (x != 0)
        product = product * x;
```

If statement with two alternatives

```
if  (rest_heart_rate > 75)
        printf("Keep up your exercise program!\n");
else
        printf("Your hear is doing well!\n");
```

Example – quadratic equation which prints roots only if they are real.

```c
#include<stdio.h>
#include<math.h>
//Quad1 - finds the roots of the quadratic equation
//         if the roots are real
int main()
{
    double a, b, c;
    double discr;
    double root1, root2;
    printf("Enter three coefficients... ");
    scanf_s("%lf%lf%lf", &a, &b, &c);
    discr = b*b - 4*a*c;
    if(discr < 0)
       printf("discriminant is negative\n");
    else
    {
        root1 = (-b + sqrt(discr))/(2*a);
        root2 = (-b - sqrt(discr))/(2*a);
        printf("roots are %lf and %lf \n", root1, root2);
    }
}
```

If statements can be nested indefinitely.

```
if (x > 0)
    num_pos = num_pos + 1
else
    if (x < 0)
        num_neg = num_neg + 1
    else   /* x equals 0 */
        num_zero = num_zero + 1
```

Example:
Quadratic equation with three cases

```c
#include<stdio.h>
#include<math.h>
//Quad2 - finds the roots of the quadratic equation
int main()
{
    double a, b, c;
    double discr;
    double root1, root2;
    double real, imag;
    printf("Enter three coefficients... ");
    scanf_s("%lf%lf%lf", &a, &b, &c);
    discr = b*b - 4*a*c;
    if(discr > 0)
    {
        printf("Roots are positive and real. \n");
        root1 = (-b + sqrt(discr))/(2*a);
        root2 = (-b - sqrt(discr))/(2*a);
        printf("roots are %lf and %lf \n", root1, root2);
    }
    else if(discr == 0)
    {
        printf("Roots are real and equal. \n");
        root1 = -b/(2*a);
        printf("roots are %lf and %lf \n", root1, root1);
    }
    else
    {
        printf("Roots are complex. \n");
        real = -b/(2*a);
        imag = (sqrt(-discr))/(2*a);
        printf("roots are %lf +/- i%lf \n", real, imag);
    }
}
```

Common programming problems with IF and logical expressions:
1. Be very careful not to confuse = with ==
The following statement is legal and gives no warnings:

```
    else if(discr = 0)
    {
        printf("Roots are real and equal. \n");
        root1 = -b/(2*a);
        printf("roots are %lf and %lf \n", root1, root1);
    }
```

The logical expression uses = instead of == so it assigns discr to 0. It then checks to see if the result is true or false. A 0 is false so this expression will always be false.

2. Remember that the logical operators need two arguments (except for !).

```
if(5 < x < 10)
```

is incorrect. For example if x = 6 this will ask is 5 < 6 which is true but that's a one so it will then ask if 1 < 10 which is true.
You should write this as:

```
if(5 < x && x < 10)
```

3. Remember that C ignores spaces which can lead to a dangling else

```
if(x < 5)
   printf("%lf", x);
else
   if(x < 10)
       printf("%lf", x – 5);
else
   printf("%lf", x + 10);
```

At first glance this looks like the second else belongs to the first if because of the spacing. But it, in fact, belongs to the second if. This would be more clear if we used braces.

```
if(x < 5)
   printf("%lf", x);
else
  {
   if(x < 10)
       printf("%lf", x – 5);
   else
       printf("%lf", x + 10);
  }
```

The switch statement

- also used to select one of several alternatives
- useful when the selection is based on the value of
  - a single variable
  - or a simple expression

- values may of type int or char
  - not double

```
switch (controlling expression) {
        label set1
                statements1
                break;
        label set2
                statements2
                break;
                .
                .
                .
        label setn
                statementsn
                break;
```

```c
/*
 * Reads serial number and displays class of ship
 */

#include <stdio.h>

int
main(void)
{
    char class;     /* input - character indicating class of ship */

    /* Read first character of serial number */
    printf("Enter ship serial number> ");
    scanf("%c", &class);         /* scan first letter */

    /* Display first character followed by ship class */
    printf("Ship class is %c: ", class);
    switch (class) {
    case 'B':
    case 'b':
            printf("Battleship\n");
            break;
    case 'C':
    case 'c':
            printf("Cruiser\n");
            break;
    case 'D':
    case 'd':
            printf("Destroyer\n");
            break;
    case 'F':
    case 'f':
            printf("Frigate\n");
            break;
    default:
            printf("Unknown\n");
    }

    return (0);
}
```

Engr 123 **August 28, 2019**
IF/ELSE

Write a console application that prompts the user for a value for x (a double). Print the corresponding value of *y* according to the following table.

$x < 0$	y = absolute value of *x*
$0 \le x < 12$	$y = x^2$
$12 \le x \le 50$	$y = (x - 12)^2$
$x > 50$	$y = \sqrt{x}$

Turn in a printed copy of your source file. Include your name, the date, and the assignment title (IF/ELSE).