BIRZEIT UNIVERSITY

Selection Structures: if and switch Statements

Comp230

Control Structure

- Three kinds of control structures
- Sequence structure
  - Programs executed sequentially by default
  - Statements executed in order
- Selection structures
  - If
  - if…else
  - switch
- Repetition structures
  - While
  - do…while
  - for

Control Structure

- Before, let us study:

1. Relational and equality operators
2. Logical Operators

Relational and equality operators

- Four different forms:

1. Variable relational-operator Variable
2. Variable relational-operator Constant
3. Variable equality-operator Variable
4. Variable equality-operator Constant

Note: You can use an expression instead of the variable or constant

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	equality
!=	not equal to	equality

Logical Operators

- Three types of logical operators:

Operator	Meaning
&&	and
||	or
!	not

Operator Precedence

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

Example

- double x=3.0 , y=4.0, z=2.0;
- int flag=0;
- What is the value after applying the following expression:

  ! flag -> !0 is 1 (true)

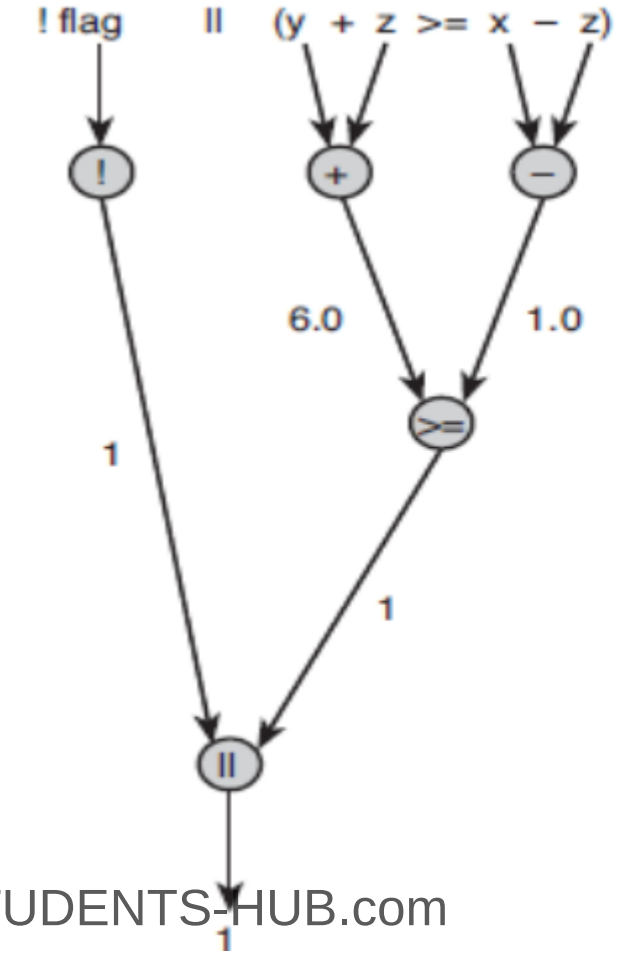
  x + y / z <= 3.5 -> 5.0 <= 3.5 is 0 (false)

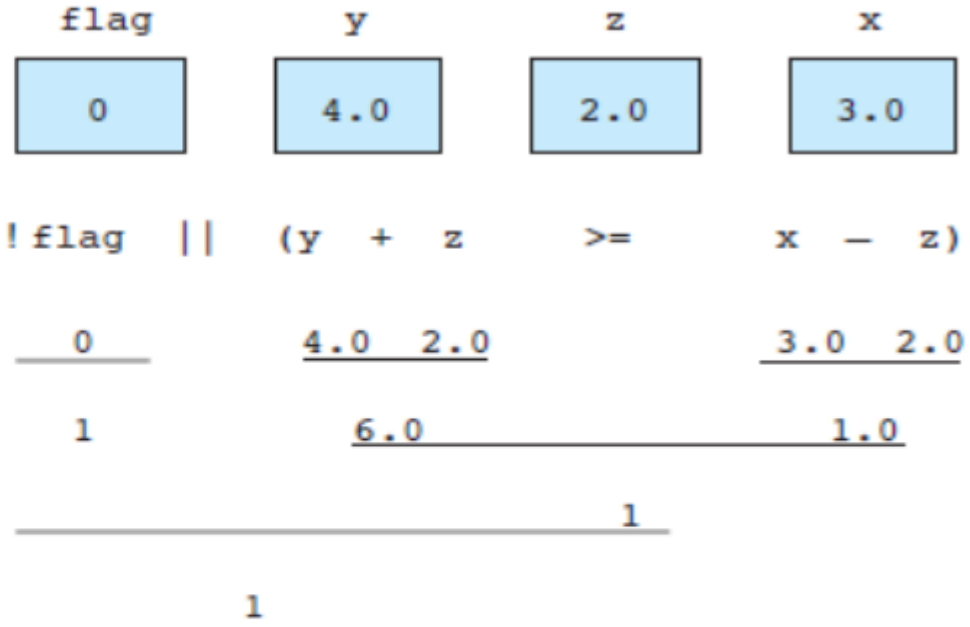
  ! flag || (y + z >= x - z) -> 1 || 1 is 1 (true)

  !(flag || (y + z >= x - z)) -> !(0 || 1) is 0 (false)

Example

- Evaluation for !flag II (y + z >= x - z)





Example: How to convert an English condition into a logical expression

- double x = 3.0 , y = 4.0 , z = 2.0 ;

English Condition	Logical Expression	Evaluation
x and y are greater than z	x>z && y>z	1 && 1 is 1 (true)
x is equal to 1.0 or 3.0	x==1.0 || x==3.0	0 || 1 is 1 (true)
x is in the range z to y, inclusive	z<=x && x<=y	1 && 1 is 1 (true)
x is outside the range z to y	!(z<=x && x<=y) z>x || x>y	!(1 && 1) is 0 (false) 0 || 0 is 0 (false)

Example: Comparing Characters

Expression	Value
‘9’ >= ‘0’	1(true)
‘a’ < ‘e’	1(true)
‘B’ <= ‘A’	0(false)
‘Z’ == ‘z’	0(false)
‘a’ <= ‘A’	system dependent (**false for ASCII**)
‘a’ <= ch && ch <= ‘z’	1(true) if ch is a lowercase letter

Logical Assignment

- Example:

```
#include <stdio.h>
int main()
{
    int age, senior_citizen;
    scanf("%d", &age);
    senior_citizen = (age >= 65);
    printf("senior_citizen =  %d",senior_citizen );
    return 0;
}
```

If Statement

- If statement with **one alternative**

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

- If statement with **two alternatives**

```
if (rest_heart_rate >56)
    printf("Your heart is in execellent health!\n");
else
    printf("Keep up your exercise program!\n");
```

if Statements with Compound Statements

```
if (condition)
{
   true task
}
Else
{
   false task
}
```

Examples

- Write a complete c program to find weather a given integer is odd or even.

```
#include <stdio.h>
int main()
{
    int number;
    printf("Please enter a number");
    scanf("%d", &number);
    if (number%2==0)
        printf("Even Integer");
    else
        printf("Odd Integer");
    return 0;
}
```

Examples

- Write a complete c program to find weather a given integer is divisible by three.

```
#include <stdio.h>
int main()
{
    int number;
    printf("Please enter a number");
    scanf("%d", &number);
    if (number%3==0)
       printf("Divisible by three");
    else
       printf("Does not divisible by three");
    return 0;
}
```

Switch X and Y example

```
if (x > y) {                    /* Switch x and y */
        temp = x;               /* Store old x in temp */
        x = y;                  /* Store old y in x */
        y = temp;               /* Store old x in y */
}
```

Multiple-Alternative Decisions

```
#include <stdio.h>
int main()
{
    int number;
    printf("Please enter a number");
    scanf("%d", &number);
    if (number>0)
        printf("Positive");
    else if (number<0)
        printf("Negative");
    else
        printf("Zero");
    return 0;
}
```

Example (if-else)

```c
#include <stdio.h>
int main()
{
    int x=0;
    if (x==1)
    {
        printf ("hello");
        printf  ("welcome");
    }
    else
    printf ("hi");
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int x=0;
    if (x==0)
    {
        printf ("hello");
        printf  ("welcome");
    }
    else
    printf ("hi");
    return 0;
}
```

Example

```c
#include <stdio.h>
int main()
{
    int x=5;
    if (x=0)
      printf ("hello");
    printf  ("welcome");
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int y=0;
    if (y)
      printf ("hello");
    printf  ("welcome");
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int y=8;
    if (y)
      printf ("hello");
    printf  ("welcome");
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int y=8,x=0;
    if (y || x)
      printf ("hello");
    printf  ("welcome");
    return 0;
}
```

Example (if, if-else)

```c
#include <stdio.h>
int main()
{
    int x=0;
    if (x==0)
    {
        printf ("hello");
        printf  ("welcome");
    }
    else
    {
        printf ("hi");
        printf  ("hi3");
    }
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int x=5;
    if (x<0)
      printf ("hello");
    printf  ("welcome");

    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int x=5;
    if (x>0)
      printf ("hello");
    printf  ("welcome");
    return 0;
}
```

Let us review the concepts:

- 1. If grade has the value of 60 , what will the following code display?

```
If (grade >= 60 )
   printf ("Passed");

   a.  nothing.
   b.  60
   c. Passed
   d. printf("Passed“);
```

Extra Exercises

- 2. What will be the value of i after the C statements at the right have been executed?

```
a.    5                    i = 3;
b.    6                    j = 10;
c.    8                    if ((3 * i) < j)
d.   10                          i = i + 2;
e.   15                    i = i + 3;
```

- 3. What is displayed by the C statements at the right if the value input is 3?

```
a.   Equal            scanf("%d", &n);
b.   Less             if (n = 5)
c.   Greater                printf("Equal\n");
d.   no output        else if (n < 5)
                            printf("Less\n");
                      else
                            printf("Greater\n");
```

The switch Statement

- The switch statement selection is based on the value of a single variable or of a simple expression.
- Expression may be of type int or char, but not of type double or string.
- The multiple selection mechanism in C is the **switch** statement.

The switch Statement

- Before,

  let us Recall:

  1. Multiple Selection with if
  2. Multiple Selection with if-else

Multiple Selection with if

```
if (day == 0 ) {
    printf ("Sunday") ;
}
if (day == 1 ) {
    printf ("Monday") ;
}
if (day == 2) {
    printf ("Tuesday") ;
}
if (day == 3) {
    printf ("Wednesday") ;
}
```

```
if (day == 4) {
    printf ("Thursday") ;
}
if (day == 5) {
    printf ("Friday") ;
}
if (day == 6) {
    printf ("Saturday") ;
}
if ((day < 0) || (day > 6)) {
    printf("Error - invalid day.\n")
;
}
```

Multiple Selection with if-else

```
if (day == 0 ) {
    printf ("Sunday") ;
} else if (day == 1 ) {
    printf ("Monday") ;
} else if (day == 2) {
    printf ("Tuesday") ;
} else if (day == 3) {
    printf ("Wednesday") ;
} else if (day == 4) {
    printf ("Thursday") ;
} else if (day == 5) {
    printf ("Friday") ;
} else if (day = 6) {
    printf ("Saturday") ;
} else {
    printf ("Error - invalid day.\n") ;
}
```

- This if-else structure is more efficient than the corresponding if structure. Why?

The switch Multiple-Selection Structure

```
switch ( integer expression )
{
   case constant1 :
         statement(s)
         break ;
   case constant2 :
         statement(s)
         break ;
         . . .
   default: :
         statement(s)
         break ;
}
```

switch Statement Details

- The last statement of each case in the switch should almost always be a break.
- The break causes program control to jump to the closing brace of the switch structure.
- Without the break, the code flows into the next case. This is almost never what you want.
- A switch statement will compile without a default case, but always consider using one.

The switch Multiple-Selection Structure

```
switch ( day )
{
      case 0:  printf ("Sunday\n") ;
               break ;
      case 1:  printf ("Monday\n") ;
               break ;
      case 2:  printf ("Tuesday\n") ;
               break ;
      case 3:  printf ("Wednesday\n") ;
               break ;
      case 4:  printf ("Thursday\n") ;
               break ;
      case 5:  printf (“Friday\n”) ;
               break ;
      case 6:  printf ("Saturday\n") ;
               break ;
      default:  printf ("Error -- invalid day.\n") ;
               break ;
}
```

Why Use a switch Statement?

- A nested if-else structure is just as efficient as a switch statement.
- However, a switch statement may be easier to read.
- Also, it is easier to add new cases to a switch statement than to a nested if-else structure.

Common Programming Errors

```
if( 0 <= x <= 4)
        printf("Condition is true\n" );
```

Instead, use

```
        if( 0 <= x && x <= 4)
```

The following always prints the same thing:

```
   if ( x = 10 )
        printf( " x is 10\n" );
```

Common Programming Errors

```
If (x = 10)
        printf(" x is 10');
```

" instead of ‘

```
If (x = 10)
        printf(" x is 10")
```

semicolon

```
If (x = 10)
        printf(" x is 10'
```

printf(" x is 10 ");

Example (Creating Menus)

```
switch( choice )
{
 case 1: printf( “Do edit\n” );
            break;
 case 2: printf( “Do delete\n” );
            break;
 case 3: printf( “Done\n” );
            break;
 default: printf( “Invalid choice!\n” );
             break;
 }
```

Example (More Practice)

- Write a C program which takes the 3 sides of a triangle, and print whether the triangle is an equilateral, isosceles or scalene triangle. Your program should include at least one function called triangle_type, this function takes the sides of the triangle and return 1 if the triangle is equilateral, 2 if the triangle is scalene and 3 for isosceles triangle.

NOTE: Your triangle should be satisfied these conditions side 1 + side 2 > side 3

side 1 + side 3 > side 2

side 2 + side 3 > side 1

Try these sides: 3 4 5

1 1 1

3 3 1

Example (More Practice)

- Write a C program which display color name based on first character of color (small or capital letters).

Note: Your program should work with the following colors: white , red and green

Example (More Practice)

- Write a C program which takes a character as input from the user. Check whether the character is an alphabet or not.