

CS19001: Programming and Data Structures Laboratory

Soumyajit Dey, Aritra Hazra;
CSE, IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Autumn2018/CS19101_PDS-Lab_Autumn2018.html

13-Aug-2018

The If-else syntax

if (boolean combination of relational expressions)

```
{ ← scope of if  
  program statement(s)  
}
```

else

```
{ ← scope of else  
  program statement(s)  
}
```

Nested If-else

```
if (condition1)
{
    if (condition2)
    {
        statement(s);
    }
    else
    { /* execute when
        (condition1 && !condition2)==TRUE */
        statement(s);
    }
}
else
{
    statement(s);
}
```

Compute $|xy|$

```
if ((x >= 0) && (y >= 0)) || ((x < 0) && (y < 0))
    z = x * y;
else
    z = -x * y;
```

- Complex conditions (error prone and non-intuitive in some cases)

Compute $|xy|$

```
if (x>=0)
{
    if (y>=0) z = x*y; else z = -(x*y);
}
else
{
    if (y>=0) z = -(x*y); else z = x*y;
}
```

- This is how we think, create sub cases based on simple conditions

Repeated If-else

```
if(condition1){
    statement(s); /* go to college */
}
else{
    if(condition2){
        statement(s); /* otherwise watch movie*/
    }
    else{
        if(condition3){
            statement(s); /* otherwise, sleep*/
        }
        else{
            -----/*thinking with elimination*/
        }
    }
} /*nesting of conditions is only in else*/
```

Repeated If-else has a simpler syntax

```
if(Condition 1){
    Block 1
}
else if(Condition 2){
    /*else if = same as saying otherwise */
    Block 2
} -----
else if(Condition n){
    Block n
}
else{
    Block n+1
}
/*need not manage complex hierarchy of braces*/
```

Implementation of the assignment $y = |x|$

```
scanf ("%d" ,&x);  
if (x == 0)  
    y = 0;  
else if (x > 0)  
    y = x;  
else y = -x;
```


Multiway (> 2) Program flow

- Still now, a program had atmost two possible execution paths at any point of time
- We simply tested boolean conditions
- Similarly, we can try matching an expression with possible set of values it may assume
- The set of possible values have to be known while writing the program

The switch statement

```
switch (E) {
    case val1 :
        Block 1 /*Execute if E = val1*/
        break;
    case val2 :
        Block 2
        break;
    .....
    case valn :
        Block n
        break;
    default   :
        /* Execute if E is equal to none */
        Block n+1
}
```

The **break** in switch

- In a switch case, once a match is found, further comparisons are disabled.
- But all following statements before the closing brace are executed one by one.
- The **break** in switch forces exit from switch

Programming Assignments

Complete and submit during lab

Assignment 1

Read from user input the real coefficients a , b , c for the quadratic equation $ax^2 + bx + c = 0$. Print out the roots of the equation in all three possible cases (real, imaginary and complex).

Assignment 2

The *distance* between two integers can be calculated by subtracting the smaller number from the bigger number.

Ex: Let two numbers be -27 and 16 .

Then, distance = $16 - (-27) = 43$, since $16 > -27$.

Write a C program to perform the following actions:

- Read from user input two integers within -50 to 50 .
- If the input numbers are not within the mentioned range $[-50, 50]$, print a message: **Error: Out-of-Range!**
- Calculate the distance.
- Print the result both in Numbers and English Words.

Output: $\langle 43, \text{Forty Three} \rangle$

Assignment 3

Write a C-program to perform the following:

- Read from user input four integers m, n, o, p .
- If m is not smaller than n and o is not smaller than p , output the message "Inputs are not Ok"
- Otherwise, consider the following arithmetic

$$[m, n] + [o, p] = [m + o, n + p]$$

$$[m, n] - [o, p] = [m - p, n - o]$$

- Compute and output the above two quantities.

Think (the logic!): What is $[m, n] * [o, p]$ and $[m, n]/[o, p]$?

Reference: https://en.wikipedia.org/wiki/Interval_arithmetic

Thank You