

C

Control Structures

Patrick Stiller

October 26, 2014

Overview

Introduction

- Advantages

- An example

Boolean Expressions

- comparative operators

- Boolean operators

case differentiation

- if then statement

- if-then-else

- switch-case

Loops

- for-loop

- while-loop

- do-while-loop



Advantages

- ▶ your program is able to make decisions
- ▶ multiply call of program-statements
- ▶ more capabilities

An example

An good example is a division. Before you divide dividende and divisor you will check that the divisor is not zero. With controll structures you can realize this decision.

comperative operators

Operator	meaning
<	smaller than
<=	smaller than or equal
>	greater than
>=	greater than or equal
==	equal
!=	unequal

Some examples

comparison	result
$5 \geq 6$	false
$1.7 < 1.8$	true
$4 + 2 == 5$	false
$2 * 4 != 7$	true

The operators

A boolean expression is an expression that has logical operators operating on boolean variables. A boolean expression evaluates to either **true** or **false**.

Operator	meaning
&&	AND
	OR
!	NOT

truth table of boolean operators

AND & OR operator

A	B	A && B	A B
false	false	false	false
false	true	false	true
true	false	false	true
true	true	true	true

Negation

A	!A
true	false
false	true

Examples for Boolean expressions

x	y	boolean expression	result
1	-1	$x \leq y \parallel y \geq 0$	false
0	0	$x > -2 \ \&\& \ y == 0$	true
-1	0	$x \ \&\& \ !y$	true
0	1	$!(x+1) \parallel y - 1 > 0$	false



If Then statement

```
1  #include <stdio.h>
2
3  int main (int argc, char *argv[]){
4      int x = 1;
5      if(x==1)//boolean statement
6      {
7          // if boolean statement is true
8          printf("x = 1 \n");
9      }
10     return 0;
11 }
```

The program will print "x = 1"



If Then Else statement

```
1  #include <stdio.h>
2  int main (int argc, char *argv[]){
3      int x = 0;
4      if(x==1)//boolean statement
5      {
6          // if boolean statement is true
7          printf("x = 1 \n");
8      }
9      else
10     {
11         // if boolean statement is false
12         printf(" x != 1");
13     }
14     return 0;
15 }
```

The program will print "x != 1"

switch-case

A switch statement allows a variable to be tested for equality against a list of values.

```
1  #include <stdio.h>
2  int main (int argc, char *argv[]){
3      switch(expression){
4          case constant-expression :
5                                  statement(s);
6                                  break; /* optional */
7          case constant-expression :
8                                  statement(s);
9                                  break; /* optional */
10     /* you can have any number of case statements */
11     default : /* Optional */
12               statement(s);
13     }
14     return 0;
15 }
```

for-loop

The for loop loops from one number to another number and increases by a specified value each time.

The structure

```
1  for(Start value; end condition; increase value)
2  {
3      statement ;
4  }
```



An example

```
1  #include<stdio.h>
2
3  int main(int argc, char *argv[])
4  {
5      int i;
6      for (i = 0; i < 10; ++i)
7      {
8          printf ("Value of i : %d\n", i);
9      }
10     return 0;
11 }
```

The program will print the numbers from zero to nine.

while-loop

The while-loop is very simple. The while loop can be used if you know the condition that you loop will continue

Basic structure

```
1  while(condition)
2  {
3      //Code to execute while the condition is true
4  }
5
```

An example

```
1  #include <stdio.h>
2
3  int main(int argc, char *argv[])
4  {
5      int x = 0;  /* Don't forget to declare variables */
6
7      while ( x < 10 ) { /* While x is less than 10 */
8          printf( "%d\n", x );
9          ++x;          /* Update x so the condition
10             can be met eventually */
11      }
12      return 0;
13 }
```


do-while-loop

do-while-loops are useful for things that want to loop at least once.

Basic structure

```
1 do {  
2     //Code to execute while the condition is true  
3 } while ( condition );
```

An example

```
1  #include <stdio.h>
2
3  int main(int argc, char *argv[])
4  {
5      int x;
6
7      x = 0;
8      do {
9          /* "Hello, world!" is printed at least one time
10             even though the condition is false */
11             printf( "Hello, world!\n" );
12         } while ( x != 0 );
13     return 0;
14 }
```