GNG1106 Lab
Tutorial 2

Codeblocks

Safa Otoum
sotou070@uottawa.ca

Variable definition

A Variable names a place in memory where you store a Value of a certain Type.

You first Define a variable by giving it a name and specifying the type, and optionally an initial value

```
char x;
char y='e';
```

Initial value of x is undefined

Initial value

Name

Type is single character (char)

The compiler puts them somewhere in memory.

Symbol	Addr	Value
	0	
	1	
	2	
	3	
x	4	?
y	5	'e'
	6	
	7	
	8	
	9	
	10	
	11	
	12	

Some operations

Expressions combine Values using Operators, according to precedence.

```
1 + 2 * 2       → 1 + 4       → 5
(1 + 2) * 2     → 3 * 2       → 6
```

Symbols are evaluated to their Values before being combined.

```
int x=1;
int y=2;
x + y * y       → x + 2 * 2     → x + 4     → 1 + 4     → 5
```

Comparison operators are used to compare values.
In C, 0 means "false", and *any other value* means "true".

```
int x=4;
(x < 5)                  → (4 < 5)                → <true>
(x < 4)                  → (4 < 4)                → 0
((x < 5) || (x < 4))     → (<true> || (x < 4))    → <true>
```

Not evaluated because first clause was true

Comparison and Mathematical Operators

- == equal to
- < less than
- <= less than or equal
- > greater than
- >= greater than or equal
- != not equal
- && logical and
- || logical or
- ! logical not

```
+  plus
-  minus
*
multiplication
/  divide
%  modulo
```

Prefix and postfix

When ++ is used as prefix (like: ++A), ++A will increment the value of *A* and then return it. but, if ++ is used as postfix(like: A++), operator will return the value of A first and then only increment it.

```
#include <stdio.h>
int main(){
int c=2,d=2;
printf("%d\n",c++);          //Print  2 then, only c incremented by 1 to 3.
printf("%d",++c);            //increments 1 to c then, only c is displayed.
return 0;}
```

Output:
2
4

Cont…

- #include <stdio.h>
- int main(){
- int var=5;
- printf("%d**\n**",var++); */* 5 is displayed then, only var is increased to 6 */*
- printf("%d",++var); */* Initially, var=6 then, it is increased to 7 then, only displayed */*
- return 0; }
- Output:
- 5
- 7

The sizeof operator

- It is a unary operator which is used in finding the size of data type, constant, arrays, structure etc. For example:

```
#include <stdio.h>
int main(){
int a;
float b;
double c;
char d;
printf("Size of int=%d bytes\n",sizeof(a));
printf("Size of float=%d bytes\n",sizeof(b));
printf("Size of double=%d bytes\n",sizeof(c));
printf("Size of char=%d byte\n",sizeof(d));
return 0;
}
```

```
Output:
Size of int=4 bytes
 Size of float=4 bytes
 Size of double=8 bytes
Size of char=1 byte
```

Conditional operators

- Conditional operators are used in decision making in C programming, i.e, executes different statements according to test condition if it is either true or false.
- conditional_expression?expression1:expression2
- If the test condition is true, expression1 is returned and if false expression2 is returned.

Write a C program to print the number entered by user only if the number entered is negative.

```
#include <stdio.h>
int main(){
int num;
printf("Enter a number to check.\n");
scanf("%d",&num);
if(num<0) { /* checking whether number is less than 0 or not. */
printf("Number = %d\n",num); }
/*If test condition is true, statement above will be executed, otherwise it will not be executed */
printf("The if statement in C programming is easy.");
return 0; }
```

```
Output:
Enter a number to check.
-2
Number = -2
The if statement in C programming is easy.
```

Write a C program to relate two integers entered by user using = or > or < sign.

```
#include <stdio.h>
int main(){
int numb1, numb2;
printf("Enter two integers to check\n");
scanf("%d %d",&numb1,&numb2);
if(numb1==numb2) //checking whether two integers are equal.
printf("Result: %d = %d",numb1,numb2);
else
if(numb1>numb2) //checking whether numb1 is greater than numb2.
printf("Result: %d > %d",numb1,numb2);
else
printf("Result: %d > %d",numb2,numb1);
return 0; }
```

Output:

```
Enter two integers to check.
 5
3
Result: 5 > 3
Enter two integers to check.
 -4
 -4
Result: -4 = -4
```

Example

```
#include <stdio.h>
int main (){
int a = 100;
if( a < 20 ){
printf("a is less than 20\n" );}
else
{
printf("a is not less than 20\n" );
}
printf("value of a is : %d\n", a);
return 0;}
```

Output:

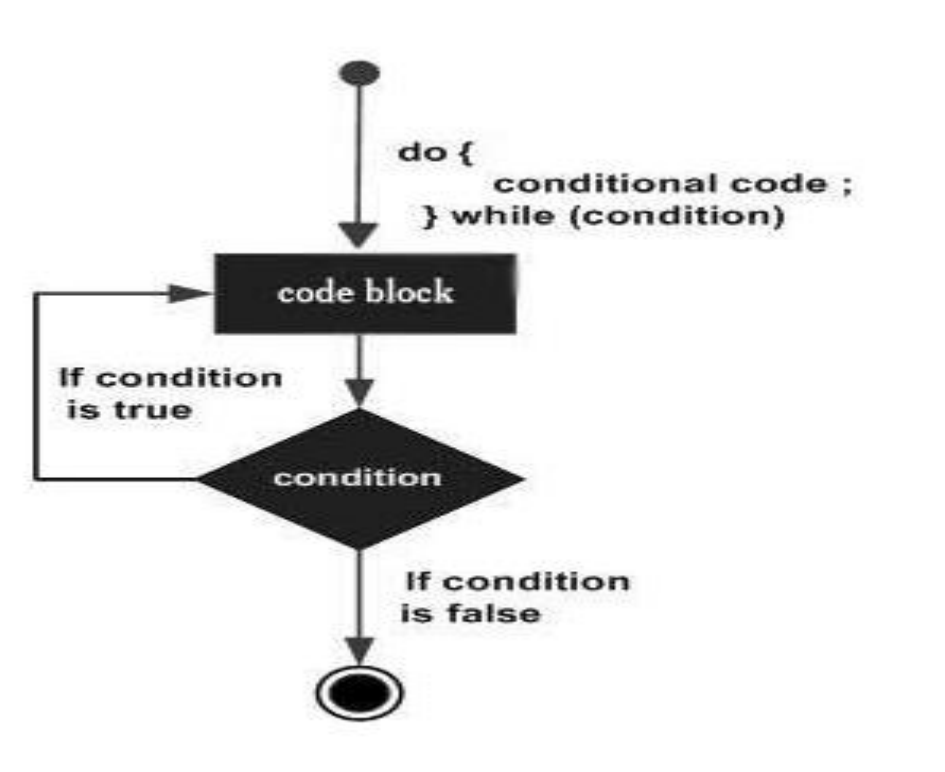
a is not less than 20; value of a is : 100

Example

```
#include <stdio.h>
int main () {
int a = 100;
if( a == 10 )
{printf("Value of a is 10\n" ); }
else if( a == 20 ) {
printf("Value of a is 20\n" ); }
else if( a == 30 ) {
printf("Value of a is 30\n" ); }
else
 {printf("None of the values is matching\n" ); }
printf("Exact value of a is: %d\n", a );
return 0;
}
```

Output:
None of the values is matching
Exact value of a is: 100

Flow Diagram of do-while



Do- while

```
#include <stdio.h>
int main () {
/* local variable definition */
int a = 10;
 /* do loop execution */
 do {
 printf("value of a: %d\n", a);
 a = a + 1;
 }while( a < 20 );
 return 0;
 }
```

Output:
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19

Difference between while and do-while loops

```
#include <stdio.h>
 int main()
{ int x = 0;
/* Don't forget to declare variables */
while ( x < 10 ) {
/* While x is less than 10 */
printf( "%d\n", x );
x++;
 /* Update x so the condition can be met eventually */ }

}
```

Output:
0 – 9 , each on a line

Continue

```
#include <stdio.h>
 int main() {
 int x;
x = 0;
do {
/* "Hello, world!" is printed at least one time even
though the condition is false */
 printf( "Hello, world!\n" );
 } while ( x != 0 );
}
```

Output:

Hello, world!

For loop

```
#include <stdio.h>
int main(){
int n, count, sum=0;
printf("Enter the value of n.\n");
scanf("%d",&n);
for(count=1;count<=n;++count) //for loop terminates if count>n
{
sum+=count; /* this statement is equivalent to sum=sum+count */
}
printf("Sum=%d",sum);
return 0;
}
```

Output

```
Enter the value of n.
19
 Sum=190
```

While loop

- // **factorial of n = 1*2*3*...*n**
-

```
#include <stdio.h>
int main(){
int number,factorial;
printf("Enter a number.\n");
scanf("%d",&number);
factorial=1;
while (number>0){ /* while loop continues util test condition number>0 is true */
factorial=factorial*number;
--number;
}
printf("Factorial=%d",factorial);
return 0;
}
```

Output:

Enter a number.

5

Factorial=120

Do-while example

```
/*C program to demonstrate the working of do...while statement*/
#include <stdio.h>
int main(){
int sum=0,num;
do
/* Codes inside the body of do...while loops are at least executed once. */
{
printf("Enter a number\n");
scanf("%d",&num);
sum+=num;
}
while(num!=0);
printf("sum=%d",sum);
return 0;
}
```

```
Enter a number
3
Enter a number
-2
Enter a number
0
sum=1
```

The End