

# **INC 141**

## **Computer Programming**

### **Lab 4**

# Learning Outcomes (Lab 4)

- Can do counter-controlled loop

# While-Loop

- **While Loop**

- The syntax of a while statement is as follows:

```
while (expression)  
    statement
```

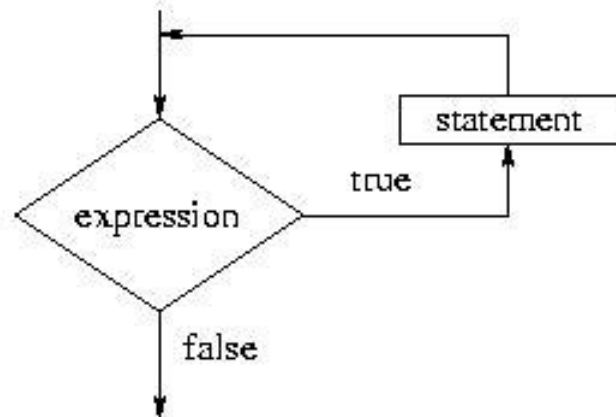
- The evaluation of the controlling expression takes place **before** each execution of the loop body.
- The loop body is executed repeatedly until the return value of the controlling expression is equal to 0.

**Note: while has the same format as if**

# Flowchart of a while Loop

- The syntax of a while loop is as follows:

```
while (expression)  
    statement
```



**Do you remember “if” flowchart?**

# Example1

```
#include <stdio.h>

main ()

{
    int i = 1;
    if(i==1)
        printf("Hello\n");
}
```

**Try changing  
this to 0**



**What happen?**

**Run normal and with debugger**

# Example1

```
#include <stdio.h>

main ()

{
    int i = 1;
    while (i==1)
        printf("Hello\n");
}
```

**Change to while**

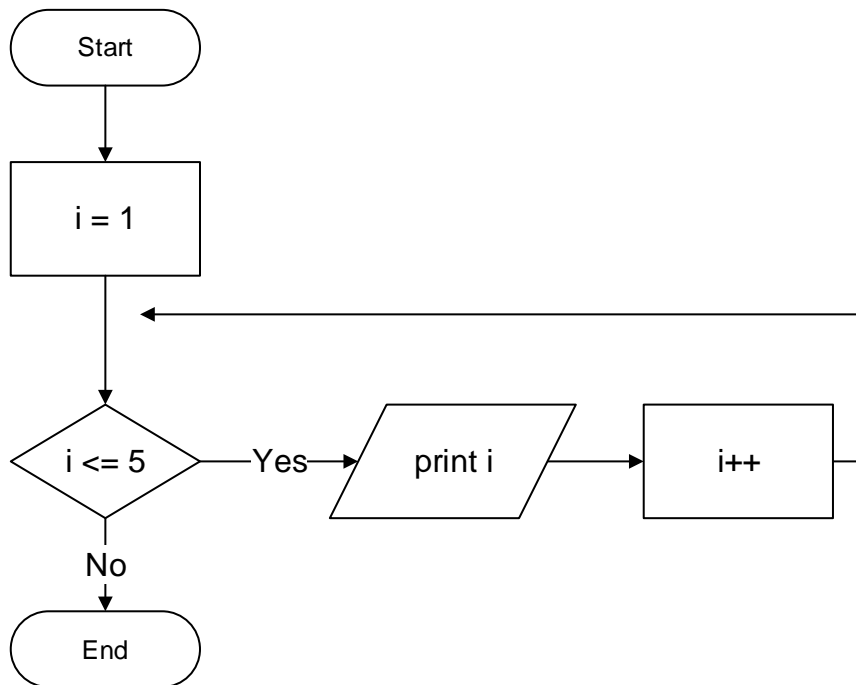
**Try changing  
this to 0**



**What happen?**

**Run normal and with debugger**

# Counter-controlled Loop



**i = counter**

# Counter-Controlled Loop

```
#include <stdio.h>

main ()

{
    int i = 1;
    while(i <= 5)
    {
        printf("%d\n", i);
        i++;
    }
}
```

**3 points**

- 1. Initial counter**
- 2. Counter condition**
- 3. Count up/down**

**Run normal and with debugger**



# Task 1

**Write a program to do these loops.**

- 1. Print numbers from 0 to 4**
- 2. Print numbers from 1 to 10**
- 3. Print numbers down from 20 to 1**

**Using while loop**

# For-Loop

**Designed for counter–controlled loop**  
**Put everything in places**

```
for(expression1; expression2; expression3)  
    statement
```

<b>Expression 1</b>	<b>Do once before entering the loop</b>
<b>Expression 2</b>	<b>Condition of the loop</b>
<b>Expression 3</b>	<b>Do every time after statement</b>

# While-For Comparison

```
#include <stdio.h>

main ()

{
    int i = 1;
    while(i <= 5)
    {
        printf("%d\n", i);
        i++;
    }
}
```

```
#include <stdio.h>

main ()

{
    int i;
    for(i=1; i <= 5; i++)
    {
        printf("%d\n", i);
    }
}
```

# Task 2

**Write a program to do these loops.**

- 1. Print numbers from 0 to 4**
- 2. Print numbers from 1 to 10**
- 3. Print numbers down from 20 to 1**

**Using for loop**

**Submit to LEB2 as figure 1**

# Do-While Loop

- **Do-While Loop**

- The syntax of a do-while statement is as follows:

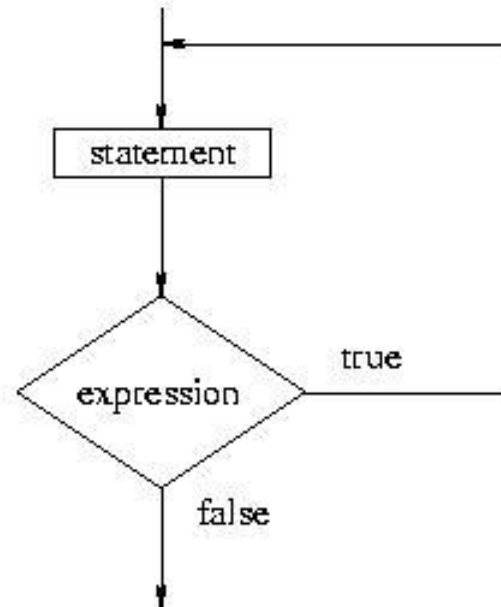
```
do  
    statement  
while (expression) ;
```

- The evaluation of the controlling expression takes place **after** each execution of the loop body.
    - The loop body is executed repeatedly until the return value of the controlling expression is equal to 0.

# Flowchart of a do-while loop

- The syntax of a do-while statement is as follows:

```
do  
    statement  
while (expression) ;
```



# While vs. Do-While

```
#include <stdio.h>

main ()

{
    int i = 1;
    while(i <= 5)
    {
        printf("%d\n",i);
        i++;
    }
}
```

```
#include <stdio.h>

main ()

{
    int i = 1;
    do
    {
        printf("%d\n",i);
        i++;
    }
    while(i <= 5);
}
```

**Run normal and with debugger**

# Task 3

**Write a program to do these loops.**

- 1. Print numbers from 0 to 4**
- 2. Print numbers from 1 to 10**
- 3. Print numbers down from 20 to 1**

**Using do-while loop**



# Task 4 (Group + LEB2)

**Write a program that calculate the sum of numbers from 1-100 and print the result on the screen.**

**Use a counter-controlled loop**

**Hint: Your program should compute in steps,**

**Round 1                       $0 + 1 = 1$**

**Round 2                       $1 \leftarrow + 2 = 3$**

**Round 3                       $3 \leftarrow + 3 = 6$**

**Round 4                       $6 \leftarrow + 4 = 10$**

**:**

**Round 100                       $? + 100 = ?$**

# Extra Task

## (Homework up on LEB2)

**Write a program that calculate the sum of only odd numbers from 1-999 and print the result on the screen.**