Arab Academy for Science and Technology and Maritime Transport

College of Engineering and Technology

Computer Engineering Department

CC112 Structured Programming

Lecture 8

Lecture 8

Repetition

Lecture Outline

- **What is a Loop?**
- **Count/Event Controlled Loops**
- **While Statement Syntax**
- **Do While Statement Syntax**

What is a loop?

- A loop is a group of instructions the computer executes repeatedly while some loop-continuation condition remains true.

Count/Event Controlled Loops

- Counter-controlled repetition is sometimes called definite repetition because we know in advance exactly how many times the loop will be executed.
- Event-controlled repetition is sometimes called indefinite repetition because it's not known in advance how many times the loop will be executed.

WHILE STATEMENT

SYNTAX

```
while  ( Expression )
{      .
         .          /*loop body */
        .
}
```

While statement

- **Every while loop will always contain three main elements:**
  - **<u>Priming</u>: initialize your variables.**
  - **<u>Testing</u>: test against some known condition.**
  - **<u>Updating</u>: update the variable that is tested.**

COUNT-CONTROLLED LOOP

```
int   count ;
count  =  4;                          /* initialize loop variable */

while (count > 0)                              /* test expression */
{
        printf(“ %d \n ”,count ) ;      /* repeated action */
        count -- ;                     /*update loop variable */
}
printf( “Done” ) ;
```

COUNT-CONTROLLED LOOP

```
#include <stdio.h>
#define MAX 10
main ()
{
        int  index =1;
        while (index <= MAX)
        {
                printf ("Index:  %d\n", index);
                index = index + 1;
        }
}
```

INFINITE LOOP

- **Infinite Loop: A loop that never ends.**

```
#include <stdio.h>
#define MAX 10
main ()
{
        int index =1;
        while (index <= MAX)
        {
        printf ("Index:  %d\n", index);
        }
}
```

Count-controlled loop

Use a while loop to read 100 grades in an exam and find their total.

```
int             thisGrade ;
int             total=0 ;
int             count ;

     count = 1 ;
while   ( count < 101 )
 {
   printf(“ Student %d  Grade : “, count );
   scanf(“ %d “ , &thisGrade ) ;
   total = total + thisGrade ;
                count++ ;
}
printf(“The total = %d  \n“ , total  ) ;
```

Event controlled loop
Flag-controlled Loops

How are they used?

- Programmer picks a value that would never be encountered for normal data
- User enters normal data and then when done, enters the unusual value
- The loop would stop when seeing the unusual value

```
total = 0;

printf(“ Grade ( -1 to stop): “ );
scanf(“ %d “ , &thisGrade ) ;

while (thisGrade != -1)
{
        total = total + thisGrade;
        printf(“ Grade ( -1 to stop): “ );
         scanf(“ %d “ , &thisGrade ) ;
}
printf(“ The total is %d \n” ,total );
```

EXAMPLE

Write a program that prints the numbers from 1 to 20 each on a separate line.

DO-WHILE STATEMENT

SYNTAX

```
do
    {
        Statements

    } while ( Expression ) ;
```

DO-WHILE STATEMENT

- The do…while repetition statement is similar to the while statement.
- In the while statement, the loop-continuation condition is tested at the beginning of the loop before the body of the loop is performed.
- The do…while statement tests the loop-continuation condition *after* the loop body is performed.
- Therefore, the loop body will be executed at least once.
- When a do…while terminates, execution continues with the statement after the while clause.

Do-While Loop vs. While Loop

- POST-TEST loop (exit-condition)
- The looping condition is tested after executing the loop body.
- Loop body is always executed at least once.

- PRE-TEST loop (entry-condition)
- The looping condition is tested before executing the loop body.
- Loop body may not be executed at all.

```
char more ;
int thisGrade , total;
 total = 0 ;
 do
     {
         printf(“ Grade : “ ) ;
         scanf(“ %d “ , &thisGrade ) ;
         total = total + thisGrade ;
         printf(“ Any more students ? (Y/N)  “ ) ;
         scanf(“ %c “ , &more ) ;
    } while ( more ==  ‘y’   ||   more ==  ‘Y’ ) ;
printf ( “ Total = %d “ , total );
```

THANK YOU