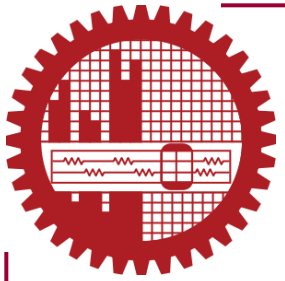

ICT 5101

Lecture 4

Dr. Hossen A Mustafa

Loop

- Executes a set of statements multiple times
- Number of execution depends on breaking expression
- { } are used to execute multiple statements
- Types
  - For
  - While
  - Do While

Loop

- Example:

```
void main(){
        int n = 1;
        printf("%d\n", n);
        n++;
        printf("%d\n", n);
        n++;
        printf("%d\n", n);
        n++;
        printf("%d\n", n);
        n++;
        printf("%d\n", n);
        n++;
        printf("%d\n", n);
        n++;
}
```

while Loop

- Example:

```
void main(){
        int n = 1;
        while(n <=10){
                printf("%d\n", n);
                n++;
        }
 }
```

- Basic Syntax:

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

for Loop

- for (*expr1; expr2; expr3)*

  statement;

- Example:

```
void main(){
        int n;
        for(n=1; n <=10; n++){
                printf("%d\n", n);
        }
 }
```

do-while Loop

- The do-while, tests expression at the bottom after making each pass through the loop body
- the statements is always executed at least once.
- Syntax

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

do-while Loop

- Example:

```
void main(){
        int n = 20;
        do{
                printf("%d\n", n);
                n++;
        }while(n <= 10);
}
```

Break and Continue

- The break statement provides an early exit from loop
- A break causes the innermost enclosing loop or switch to be exited immediately.

```
void main(){
        int n = 1;
        while(n <=10){
                printf("%d\n", n);
                n++;
                if(n == 5)
                        break;
        }
 }
```

Break and Continue

- The break statement provides an early exit from current iteration of the loop
- A continue causes skip of rest of the loop

```
void main(){
        int n = 1;
        while(n <=10){
                if(n == 5)
                        continue;
                printf("%d\n", n);
                n++;
        }
 }
```

Infinite Loop

- A loop becomes infinite when the breaking condition is never met
- The program will never terminate
- Example:

```
void main(){
        int n;
        for(n=1; n <=10; n--){
                printf("%d\n", n);
        }
}
```

Infinite Loop

```
for(;; );

while(1);

do{ }while(1);
```

- How can you go out of a infinite loop?

Assignment

- Write a program named assignment2.c
- The program should take an Integer n as input from keyboard
- The program should output 3 summations using loop:
  - Sum of all numbers between 1 and n
  - Sum of all odd numbers between 1 and n
  - Sum of all even numbers between 1 and n
- Example:
  - Input = 10
  - Output
    - SUM ALL = 55
    - SUM ODD = 25
    - SUM EVEN = 30

Homework Assignment

- Write a program named hwassignment1.c
- The program should take an Integer n as input from keyboard
- The program should output all the prime numbers between 1 and n

- Example:
  - Input = 30
  - Output: 2 3 5 7 11 13 17 19 23 29

- Deadline: 10/06/17