# C under Linux

Dr. Naeem Odat


Department of Computer and Communications Engineering

C - Looping

# C - Looping

# C - Looping

## Loops in C

- while loop
- do ... while loop
- for loop

# C - Looping

## while loop example

```
#include<stdio.h>
int main(){
    int nSum = 0, nCount = 1;
    while (nCount <= 100){
        nSum = nSum + nCount;
        nCount++;
    }
    printf(\Sum is %d\n", nSum);
    return 0;
}
```

# C - Looping

## do while loop example

```
#include<stdio.h>
int main(){
    int nSum = 0, nCount = 1;
    do{
        nSum = nSum + nCount;
        nCount++;
    } while(nCount <= 100);
    printf(\Sum is %d\n", nSum);
    return 0;
}
```

# C - Looping

## for loop example

```
#include<stdio.h>
int main(){
    int nSum = 0, nCnt = 1;
    for(nCnt = 0; nCnt <=100; nCnt++){
        nSum = nSum + nCnt;
    }
    printf(\Sum is %d\n", nSum);
    return 0;
}
```

# C - Looping

## Infinite loops - for

```
for(;;){
    statement;
    .
    .
    .
}
```

## Infinite loops - while

```
while(1){
    statement;
    .
    .
    .
}
```

# C - Looping

## The break/continue Statements

- Break causes immediate exit from a block of code.

```
#include <stdio.h>
int main(){
    int nCnt = 1;
    while(nCnt <= 10){
        if (5 == nCnt)
            break;
        printf(\%d \, nCnt);
        nCnt++;
    }
    printf("Broke at %d", nCnt);
    return 0;
}
```

# C - Looping

## The break/continue Statements

- Continue skips the remaining statements in a loop and proceeds to the next iteration.

```
#include <stdio.h>
int main(){
    int nCnt = 1;
    while(nCnt <= 10){
        if (5 == nCnt){
            nCnt++;
            continue;
        }
        printf(\%d\n", nCnt);
        nCnt++;
    }
    printf("Used continue to skip 5");
    return 0;
}
```