BINUS UNIVERSITY

People
Innovation
Excellence

Basic Programming in C

Program Control: Repetition

D4730 – Rini Wongso, S.Kom., M.T.I

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition Definition

- One or more instruction repeated for certain amount of time
- Number of repetition can be predefined (hard-coded in program) or defined later at run time
- Repetition/looping operation:
  - **for**
  - **while**
  - **do-while**

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition: FOR

- **Syntax:**

```
for(exp1; exp2; exp3) statement;
or:
for(exp1; exp2; exp3){
        statement1;
        statement2;
        .......
 }
```

exp1 : initialization
exp2 : conditional
exp3 : increment *or* decrement
exp1, exp2 and exp3 are optional

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition: FOR

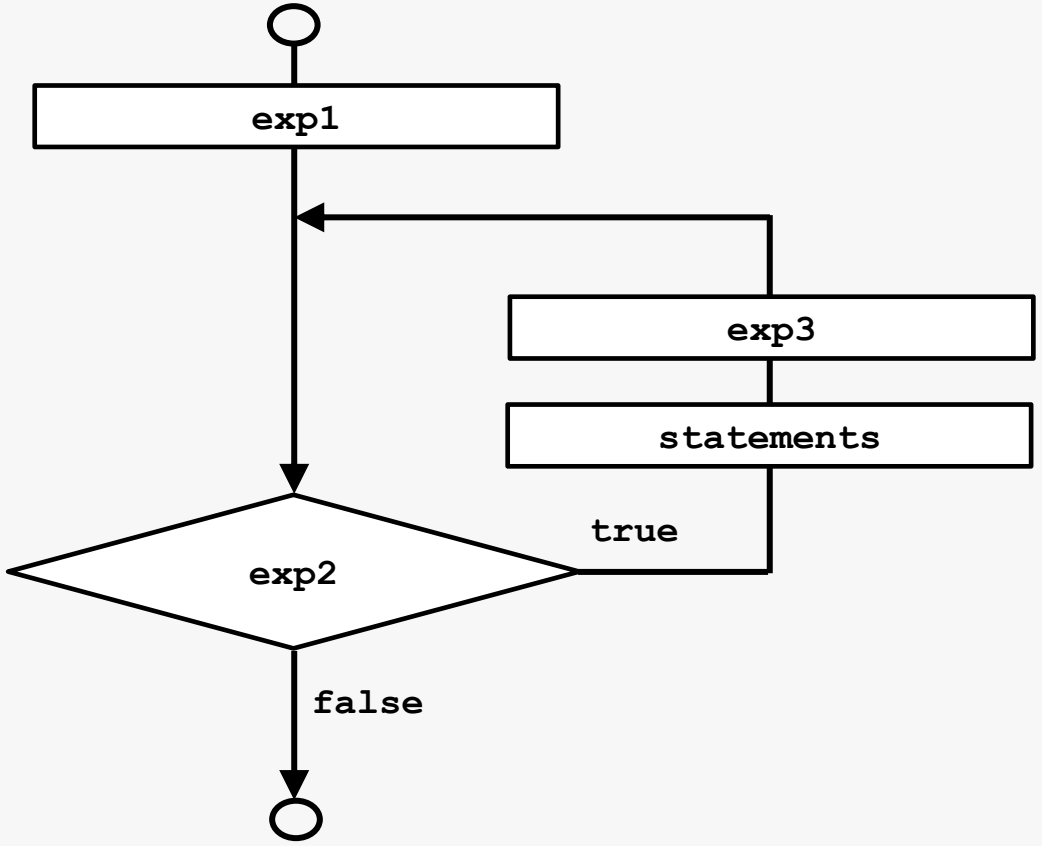
- exp1 and exp3 can consist of several expression separated with comma

- Example:

```
void displayNumber(){
    int i;
    for(i=1; i<=10; i++){
        printf("%d ", i);
    }
}
```

Repetition: FOR

- Flow Chart of **FOR** Statement

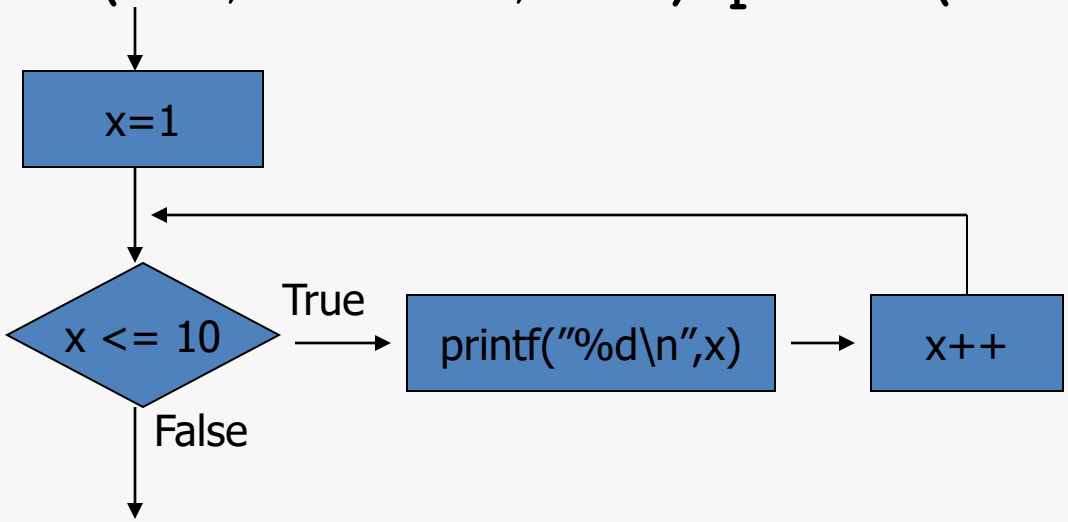


People
Innovation
Excellence

Repetition: FOR

- **Example:**

```
for (x=1; x <= 10; x++) printf("%d\n", x);
```



People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition: FOR

- **Example:**
  - Program to print out numbers from 1 to 10

```
#include<stdio.h>
int main()
{
    int x;
    for( x = 1 ;  x <= 10 ;  x++ ) printf( "%d\n", x );
    return(0);
}
```

  - Program to print out numbers from 10 to 1

```
#include<stdio.h>
int main()
{
    int x;
    for( x = 10 ;  x >= 1 ;  x-- ) printf( "%d\n", x);
    return(0);
}
```

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition: FOR

- **Infinite Loop**

  Loop with no stop condition can use "for-loop" by removing all parameters (exp1, exp2, exp3). To end the loop use **break**.

- **Nested Loop**

  Loop in a loop. The repetition operation will start from the inner side loop.

Repetition: FOR

C

```
int x, y;
for (x=1;x<=5;x++)
     for (y=5; y>=1; y--)
          printf("%d %d ",x,y);
```

C++

```
for (int x=1;x<=5;x++)
     for (int y=5; y>=1; y--)
          printf("%d %d ",x,y);
```

NESTED LOOP

Output :

1 5 1 4 1 3 .. 2 5 2 4 .. 5 1

People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition: WHILE

- **Syntax :**

  while (exp) statements;

  or:

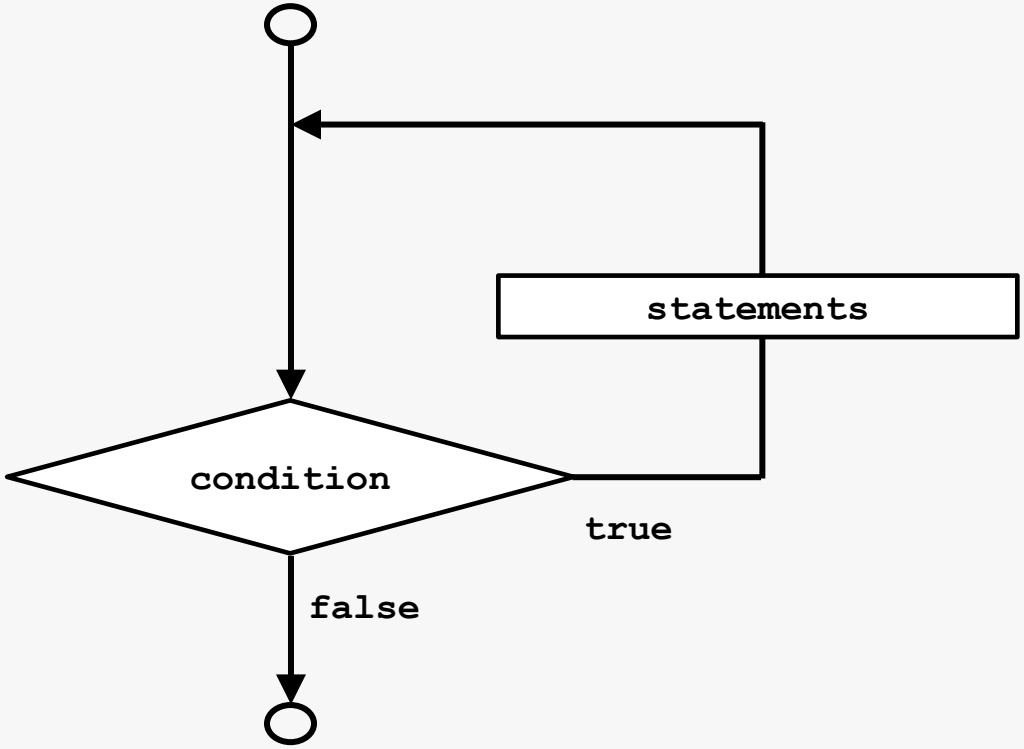
  while(exp){
  statement1;
  statement2;
  .....
  }

Example :

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

Repetition: WHILE

- Flow Chart of **WHILE** Statement



People
Innovation
Excellence

Repetition: WHILE

while (exp) statements;

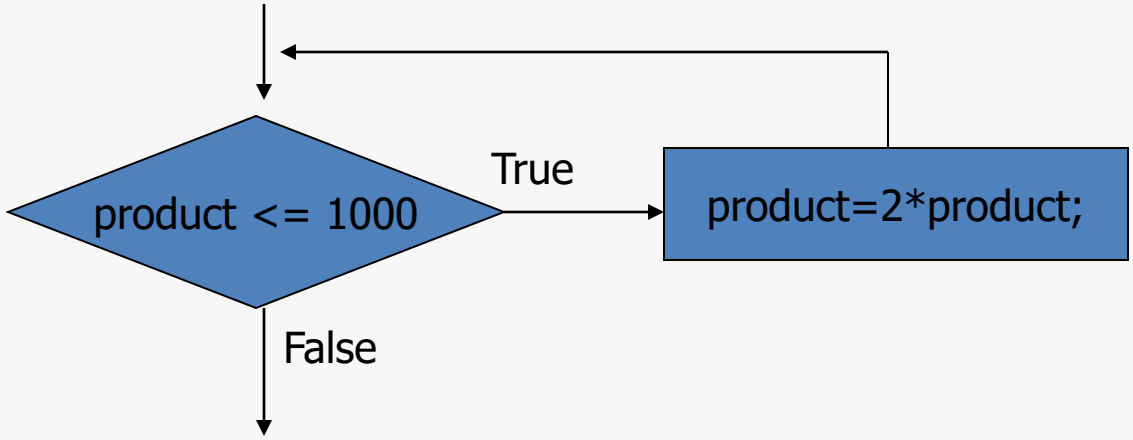
- ***exp*** is Boolean expression. It will result in true (not zero) or false (equal to zero).
- Statement will be executed while the ***exp*** is not equal to zero.
- ***exp*** evaluation is done before the statements executed.

People
Innovation
Excellence

Repetition: WHILE

- Example :

```
while(product <= 1000) product = 2*product;
```



People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

Repetition: WHILE

These code are analogous

```
#include<stdio.h>
void main() {
   int x;
   for( x = 1 ;  x <= 10 ;  x++ )
         printf( "%d\n", x );
}
```

```
#include<stdio.h>
void main() {
   int x = 1;
   while (x<=10) {
         printf( "%d\n", x );
         x++;
   }
}
```

Repetition: DO-WHILE

- **Syntax :**

```
do{
    < statements >;
} while(exp);
```

Example :

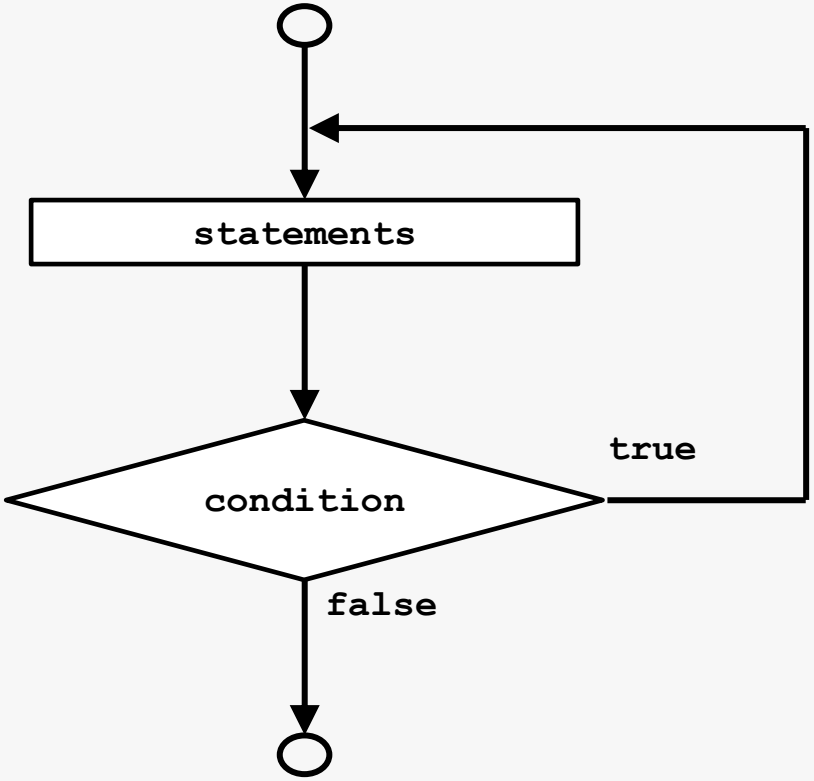
```
int counter=0;
do {
      printf( "%d  ", counter );
      ++counter;
} while (counter <= 10);
```

- Keep executing while **exp** is true
- **exp** evaluation done after executing the statement(s)

People
Innovation
Excellence

Repetition: DO-WHILE

- Flow Chart of **DO-WHILE** Statement

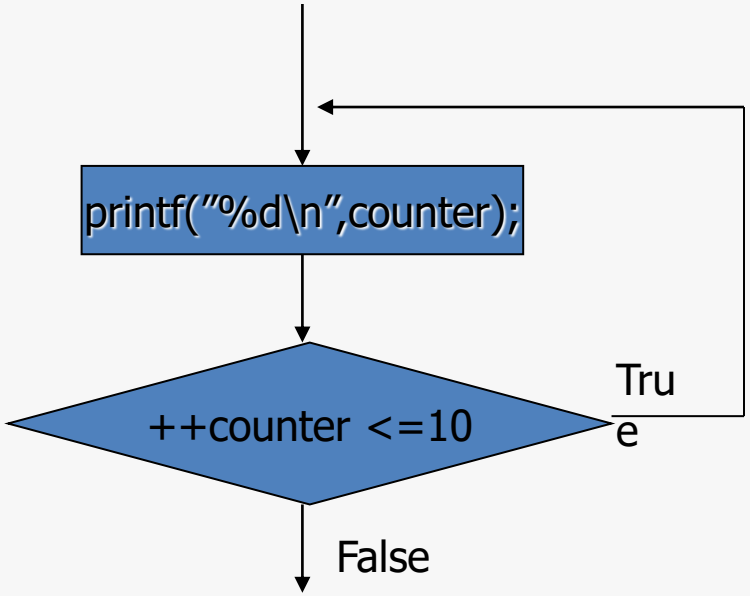


People
Innovation
Excellence

Repetition: DO-WHILE

- Example:

```
do{
        printf("%d\n",counter);
} while(++counter <=10);
```



People
Innovation
Excellence

BINUS UNIVERSITY

Repetition Operation

- In **while** operation, statement block of statements may never be executed at all **<u>if exp value is false</u>**
- In **do-while** on the other hand statement block of statements **<u>will be executed min once</u>**

People
Innovation
Excellence

Repetition Operation

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

end the loop

Break is to **force** finish the loop

People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

Break vs Continue

- **break**:
  - ending loop (for, while and do-while)
  - end the **switch** operation

- **continue**:

  skip all the rest of statements (subsequent to the skip statement) inside a repetition, and continue normally to the next loop

BINUS UNIVERSITY

People
Innovation
Excellence

Break vs Continue

- Example using `continue`:

```
#include <stdio.h>
int main() {
        int x;
        for(x=1; x<=10; x++) {
        if (x == 5) continue;
        printf("%d ", x);
        }
        return 0;
}
```

Output : 1 2 3 4 6 7 8 9 10

BINUS UNIVERSITY

People
Innovation
Excellence

Break vs Continue

- Example using `break`:

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

Output : 1 2 3 4

BINUS UNIVERSITY

People
Innovation
Excellence

Exercise

1.
```
#include <stdio.h>
int main()
{
   int x,y;
   for(x=1;x<=3;x++)
     for (y=3;y>=1;y--)
           printf("%d %d ",x,y);
    return 0;
}
```

What are their output ?

```
#include <stdio.h>
int main()
{
   int x,y;
   for(x=1;x<=3;x++);
     for (y=3;y>=1;y--)
           printf("%d %d
   ",x,y);
    return 0;
}
```

watch for the semicolon

Exercise

2. Create a program using C to display odd numbers from 11 to 188, using :
   - for
   - while
   - do-while

3. Given: 1 is Monday, 2 - Tuesday, 3 – Wednesday , and so on. Using C, create a program to show the value of each day in a week using n - input from keyboard. Example:

```
N = 3
      1   2   3
N = 7
      1   2   3   4   5   6   7
N = 10
1   2   3   4   5   6   7   1   2   3
```

People
Innovation
Excellence

BINUS UNIVERSITY

Summary

- Repetition is a condition which is one or more instruction repeated for certain amount of time
- 3 types of repetition/looping in C:
  - for
  - while
  - do-while

People
Innovation
Excellence

BINUS UNIVERSITY

References

- Paul Deitel & Harvey Deitel. (2016). C how to program : with an introduction to C++. 08. Pearson Education. Hoboken. ISBN: 9780133976892. Chapter 3 & 4
- Doing the Same Thing Over and Over: http://aelinik.free.fr/c/ch07.htm

People
Innovation
Excellence

BINUS UNIVERSITY

People
Innovation
Excellence

END