Mahidol University

ITCS113 Fundamentals of Programming

Lecture 3 - Repetition

Instructor: Asst. Prof. Dr. Akara Supratak

Contact: akara.sup@mahidol.edu

Quiz#1

- Closed-book
- 15:30 - 16:30 (60 mins)
- Questions are on MyCourse
- Scope: Lecture 1 and 2

Agenda

- Basic Loop Structure
- `while` statement
- `for` statement

Basic Loop Structure

1. **Repetition Statement**

   `while, for, do-while`

2. **Relational Expression** (i.e., condition)

   If the condition is **`True`**, do the codes in the loop

3. **Initialization Statement**

   A statement setted before the condition being evaluated

4. **Alteration**

   A statement in the repetition section of code that change the condition until it becomes **`False`**

while statement

`while` statement

flowchart

Example

Exercise

1 to 100 increased by 1

100 to 1 decreased by 2

More Exercises

- 1, 3, 5, 7, 9
- 10, 8, 6, 4, 2
- 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
- a, b, c, d, e, f, g, h, i, j, k
- 1, 2, 4, 8, 16, 32
- 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20
- 1c, 2f, 3c, 4f, 5c, 6f, 7c, 8f, 9c, 10f
- 1a, 2b, 3c, 4d, 5e, 6f, 7g, 8h, 9i, 10j

`while` statement + `scanf`

- Ex1: Receive 10 numbers and print out the accumulated sum after every input
- Ex2: Receive input numbers *until* a user input a negative number, then calculate and display the sum of the input numbers.

Ex1

Ex2

`for` statement

flowchart

`for` statement

Example

Exercise

1 to 100 increased by 1

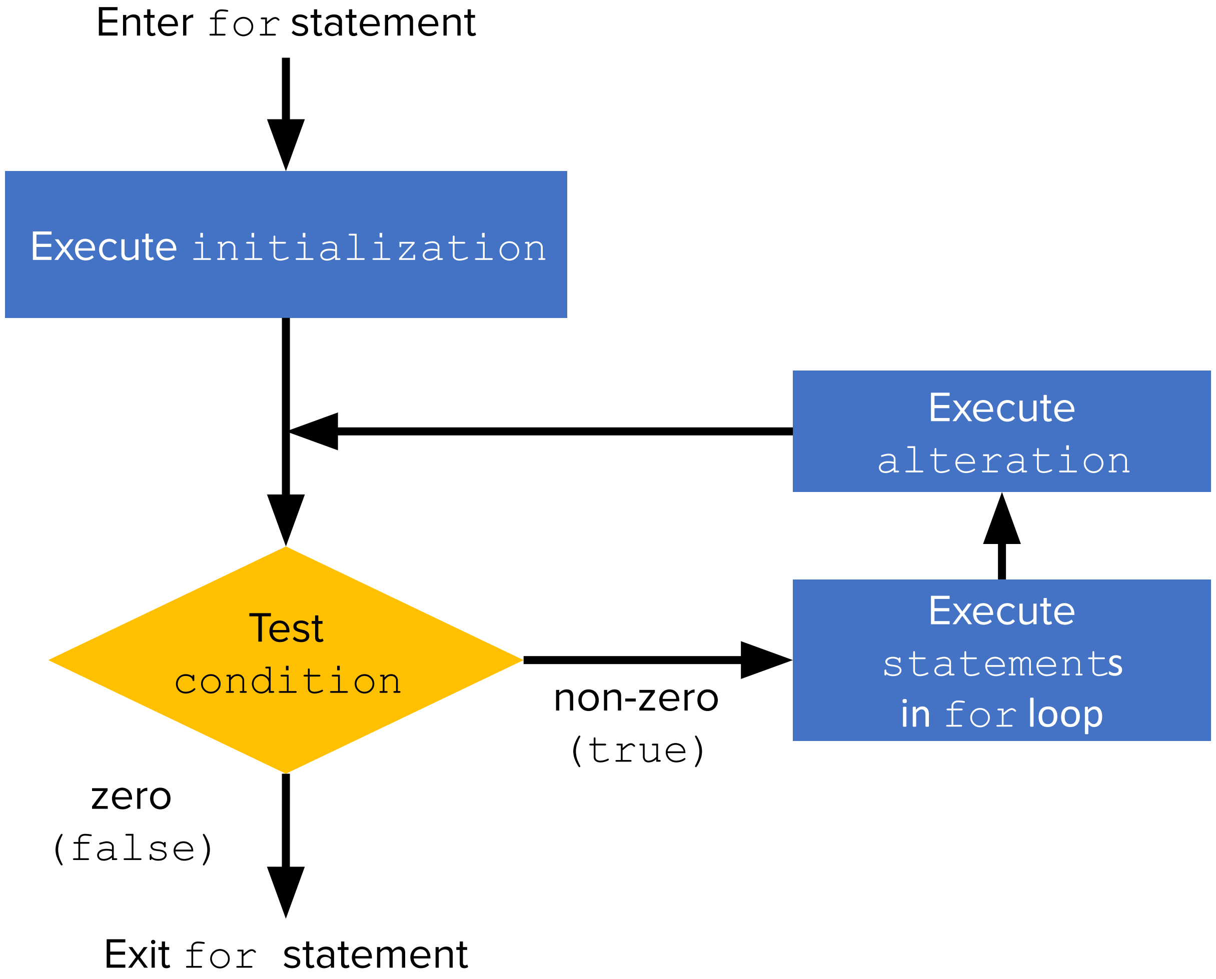
100 to 1 decreased by 2

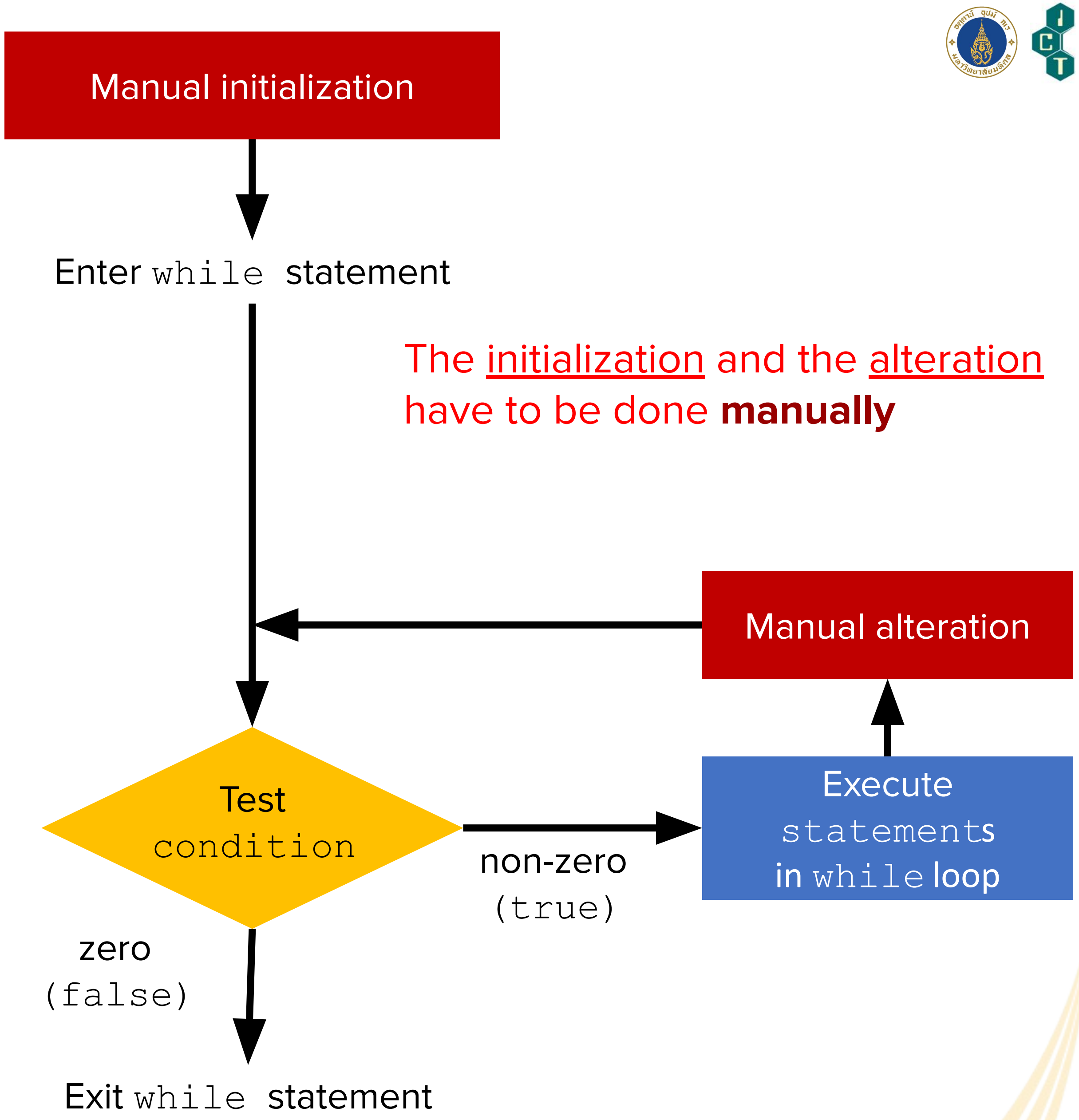
More Exercises

- `1, 3, 5, 7, 9`
- `10, 8, 6, 4, 2`
- `1, 1, 1, 1, 1, 1, 1, 1, 1, 1`
- `a, b, c, d, e, f, g, h, i, j, k`
- `1, 2, 4, 8, 16, 32`
- `1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20`
- `1c, 2f, 3c, 4f, 5c, 6f, 7c, 8f, 9c, 10f`
- `1a, 2b, 3c, 4d, 5e, 6f, 7g, 8h, 9i, 10j`

`while vs for` statement

for VS while





The initialization and the alteration have to be done **manually**

for VS while

for loop

```
int count;
for (count = 1; count<=10; count++)
    printf("%d", count);
```

while loop

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

Output

```
1 2 3 4 5 6 7 8 9 10
```

The **for** statement

- Mostly used when we KNOW in advance how many times a loop will execute

The **while** statement

- Mostly used when we DON'T KNOW the number of repetitions in advance
- But it can be used in both the situations, depending on the coding style

`for` VS `while`

- Display even numbers in the range between 2 and 100

`for` statement

`while` statement

Which one is better?

for VS while

- Display even numbers in the range between 2 and 100

for loop

```
int i;
for ( i = 2 ; i <= 100 ; i = i + 2 )
    printf("%d ", i);
```

✓

while loop

```
int i = 2;
while (i <= 100)
{
    printf("%d ", i);
    i = i + 2;
}
```

for VS while

- Compute the sum from 1 to 10

for statement

while statement

Which one is better?

for VS while

- Compute the sum from 1 to 10

for loop

```
int sum = 0, count;
for (count = 1; count <= 10; count++) {
    sum = sum + count;
}
printf("Sum=%d", sum);
```

✓

while loop

```
int sum = 0, count = 1;
while (count <= 10) {
    sum = sum + count;
    count++;
}
printf("Sum=%d", sum);
```

for VS while

- Repeatedly ask a user to input a number until the user inputs a negative number

for statement

while statement

Which one is better?

`for` VS `while`

Repeatedly ask a user to input a number *until* the user inputs a negative number

`while` loop

```
int number = 0;
while (number >= 0) {
    scanf("%d", &number);
    printf("You enter number:
    %d\n", number);
}
printf("Exit!!\n");
```

Output

```
Please enter number: 3
You enter number: 3
Please enter number: 89
You enter number: 89
Please enter number: 7
You enter number: 7
Please enter number: 1443
You enter number: 1443
Please enter number: -1
You enter number: -1
Exit!!
```

`for` VS `while`

- Repeatedly ask a user to input a number *until* the user inputs a negative number

`while` loop

```
int number = 0;
while (number >= 0) {
     scanf("%d", &number);
     printf("You enter number:
     %d\n", number);
}
printf("Exit!!\n");
```

Output

```
Please enter number: 3
You enter number: 3
Please enter number: 89
You enter number: 89
Please enter number: 7
You enter number: 7
Please enter number: 1443
You enter number: 1443
Please enter number: -1
You enter number: -1
Exit!!
```

Can we do this with **`for`** statement? Is it appropriate?

Be careful: Infinite loop

- This term refers to the looping non-stop behavior of the program
- The action occurs due to the incorrect expression or alteration

- **For example**

```
int i;
for ( i = 1 ; i > 0 ; i=i+1 )
    printf("%d ", i);
printf("\n");
```

This loop will never stop
as `i` keeps increasing

- To break it, in the terminal, press CTRL + C or press STOP in Repl.it

Lab Exercises