Tishk International University
Faculty of Applied Science
Information Technology Department

Iterative Control Structures (While & Do-While Loops)

Lecture 7

Fall 2025

Course Code: IT117

Grade 1

Islam Abdulazeez

islam.abdulaziz@tiu.edu.iq
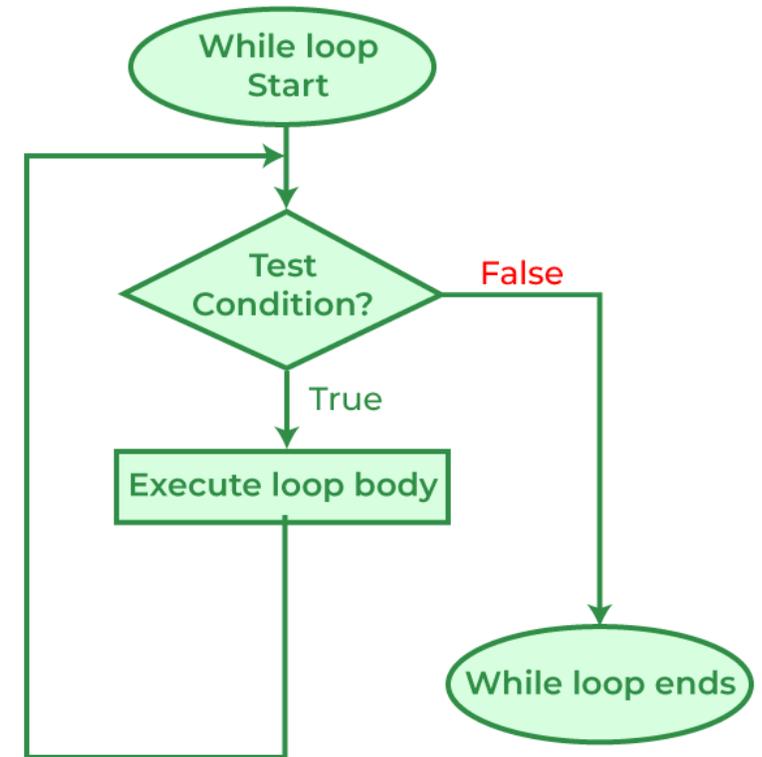
February 1, 2026

Programming I

Outlines

- ✓ Iterative Control Structures

- ✓ while Loop

- ✓ Input Validation Using while Loop

- ✓ do-while Loop

- ✓ Comparison: while vs do-while

Learning Outcomes

- **At the end of today's session, you will be able to:**

  - ✓ Define while and do-while loops

  - ✓ Explain how loop conditions control repetition

  - ✓ Apply while and do-while loops in C++ programs

  - ✓ Differentiate between while and do-while loops

  - ✓ Write programs using iterative control structures

- A **while loop** is a control structure that repeatedly executes a block of code as long as a condition is true.

- Recommended when the loop depends on user input or a condition

➢ Basic idea

  ▪ First, the condition is checked

  ▪ If the condition is true, the loop body runs

  ▪ After each run, the condition is checked again

  ▪ The loop stops when the condition becomes false

While loop Start

Test Condition? — False — True

Execute loop body

While loop ends

```
while (condition) {

    // code to be executed

}
```

```cpp
int i = 1;

while (i <= 5) {
    cout << "Programming" << endl;

}
```

Infinite Loop ✗

```
Programming
Programming
Programming
Programming
        :
```

```cpp
int i = 1;

while (i <= 5) {
    cout << "Programming" << endl;
    i++;
}
```

```
Programming
Programming
Programming
Programming
Programming
```

```cpp
int i = 4;

while (i >= 1) {
    cout << i << endl;
    i--;
}

cout << "Thank you!";
```

```
4
3
2
1
Thank you!
```

- Write C++ code that prints numbers from 1 to 10 and finds the square for each.

```cpp
int i = 1;

cout << "Number \t\t Square" << endl;

while (i <= 10) {
    cout << i << "\t\t" << i * i << endl;
    i++;
}
```

```
Number              Square
1                   1
2                   4
3                   9
4                   16
5                   25
6                   36
7                   49
8                   64
9                   81
10                  100
```

- Write a C++ program that asks the user to enter a positive number. If the number is not positive, keep asking until a positive number is entered, then show the number.

```cpp
int num;

cout << "Enter a positive number: ";
cin >> num;

while (num <= 0) {
    cout << "Please enter a positive number: ";
    cin >> num;
}

cout << "You entered a positive number: " << num;
```

```
Enter a positive number: -3
Please enter a positive number: -5
Please enter a positive number: 2
You entered a positive number: 2
```

Same logic as in slide number 10

- Write a C++ program that keeps asking the user to enter a positive number until a valid one is entered, then displays the number.

```cpp
int num;

while (true) {
    cout << "Enter a positive number: ";
    cin >> num;

    if (num > 0) {
        break;
    }
    cout << "Invalid input. ";
}
cout << "You entered a positive number: " << num;
```

```
Enter a positive number: -5
Invalid input. Enter a positive number: -2
Invalid input. Enter a positive number: 3
You entered a positive number: 3
```

- Write a C++ program that asks the user to enter a grade from 0 to 100. If the grade is outside this range, keep asking until a valid grade is entered. Then print "Passed" if the grade is 50 or higher, otherwise print "Failed".

```cpp
int grade;

cout << "Enter a grade: ";
cin >> grade;

while (grade < 0 || grade > 100) {
    cout << "Invalid grade, enter grade again: ";
    cin >> grade;
}

if (grade >= 50) {
    cout << "Passed" << endl;
}
else {
    cout << "Failed" << endl;
}
```

Q. Write a C++ program that lets the user enter grades until -1 is entered, then calculates and displays the average of the grades.

```cpp
int grade, counter = 0;
double total = 0;

cout << "Enter a grade (-1 to exit): ";
cin >> grade;

while (grade != -1) {
    total = total + grade;
    counter++;
    cout << "Enter a grade (-1 to exit): ";
    cin >> grade;
}

cout << "The average of grades is: " << total / counter;
```

```
Enter a grade (-1 to exit): 76
Enter a grade (-1 to exit): 85
Enter a grade (-1 to exit): 74
Enter a grade (-1 to exit): 91
Enter a grade (-1 to exit): -1
The average of grades is: 81.5
```

- Write a C++ program that repeatedly asks the user to enter a number and then asks whether to continue. The program should keep running while the user enters y, and stop when any other character is entered.

```cpp
int num;
char choice = 'y';

while (choice == 'y') {
    cout << "Enter a number: ";
    cin >> num;
    cout << "Do you want to continue? (y/n): ";
    cin >> choice;
}

cout << "Program ended." << endl;
```
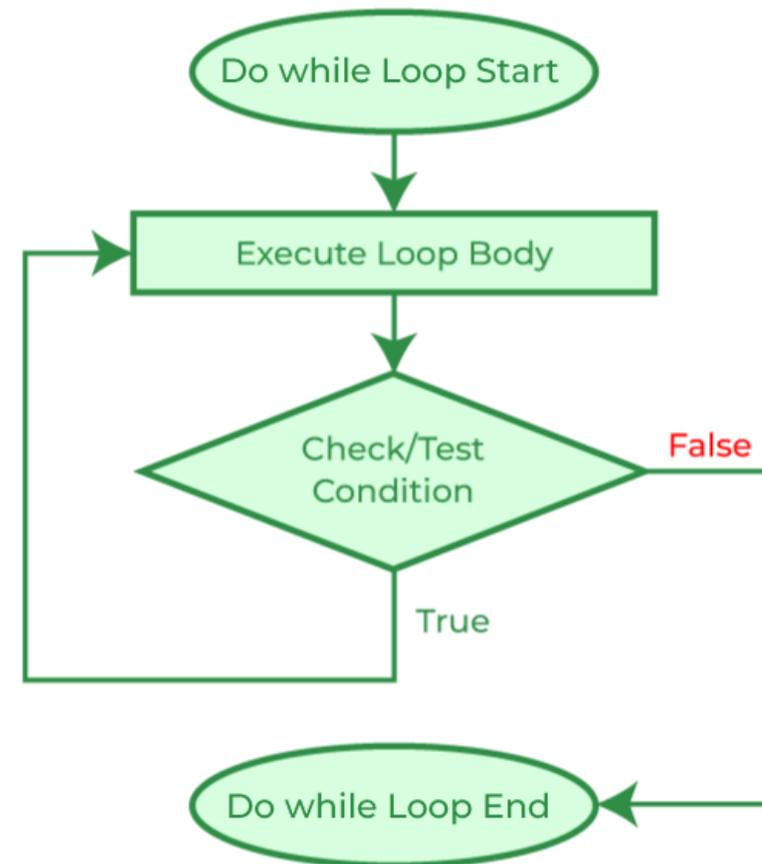
```
Enter a number: 4
Do you want to continue? (y/n): y
Enter a number: 8
Do you want to continue? (y/n): n
Program ended.
```

- A **do-while loop** in C++ is a loop that executes the code **at least once**, then checks the condition.

```
do {

    // code to be executed

} while (condition);
```
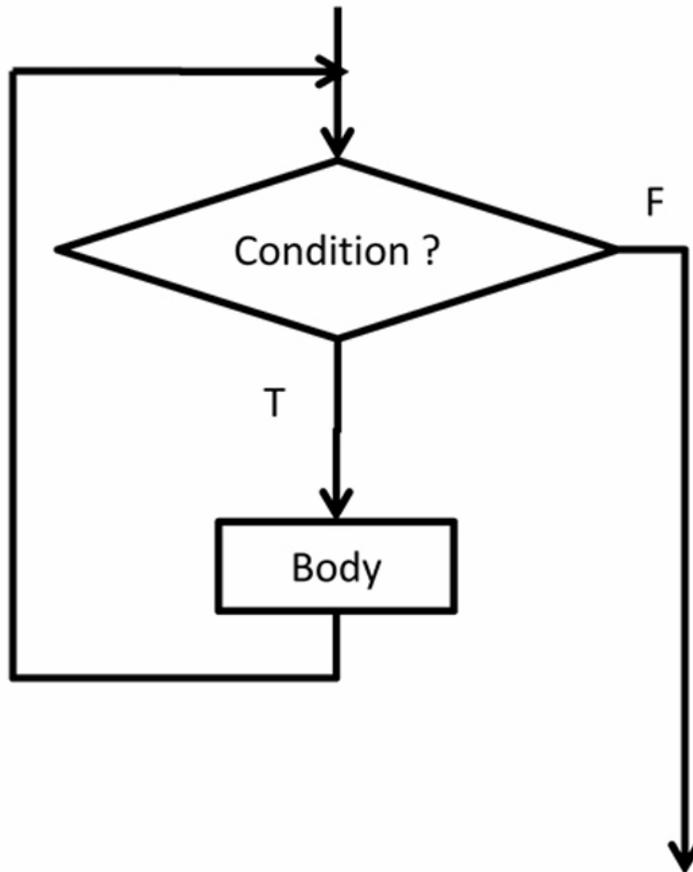
While versus Do-While Loops
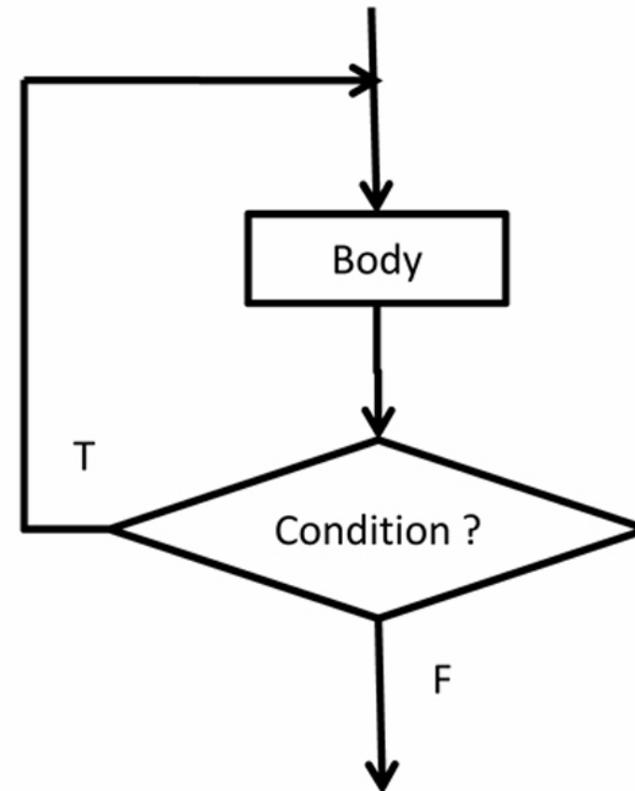
```
int b = 10;
while (b <= 5)
{
    cout << "TIU" << endl;
    b++;
}
```

0 loop

```
int b = 10;
do
{
    cout << "TIU" << endl;
    b++;
} while (b <= 5);
```

1 loop

```cpp
int i = 1;

do {
    cout << i << endl;
    i++;
} while (i <= 5);
```

```
1
2
3
4
5
```

```cpp
int number;

do {
    cout << "Enter a positive number: ";
    cin >> number;
} while (number <= 0);

cout << "Thank you for entering a positive number!";
```

```
Enter a positive number: -2
Enter a positive number: -7
Enter a positive number: 3
Thank you for entering a positive number!
```

```cpp
int number;

do {
    cout << "Enter a positive number: ";
    cin >> number;

    if (number <= 0) {
        cout << "Please enter a number greater than 0!\n";
    }
} while (number <= 0);

cout << "Thank you! You entered: " << number << endl;
```

```
Enter a positive number: -5
Please enter a number greater than 0!
Enter a positive number: 7
Thank you! You entered: 7
```

```cpp
char choice;

do {
    cout << "Press y to continue: ";
    cin >> choice;
} while (choice == 'y');

cout << "The loop has stopped!";
```

```
Press y to continue: y
Press y to continue: y
Press y to continue: y
Press y to continue: a
The loop has stopped!
```

Q. Write a C++ program that reads integers until -1 is entered, then displays how many numbers were entered

```cpp
int number;
int counter = 0;

cout << "Enter numbers (-1 to stop):\n";

do {
    cin >> number;

    if (number != -1) {
        counter++;
    }
} while (number != -1);

cout << "You entered " << counter << " numbers.\n";
```
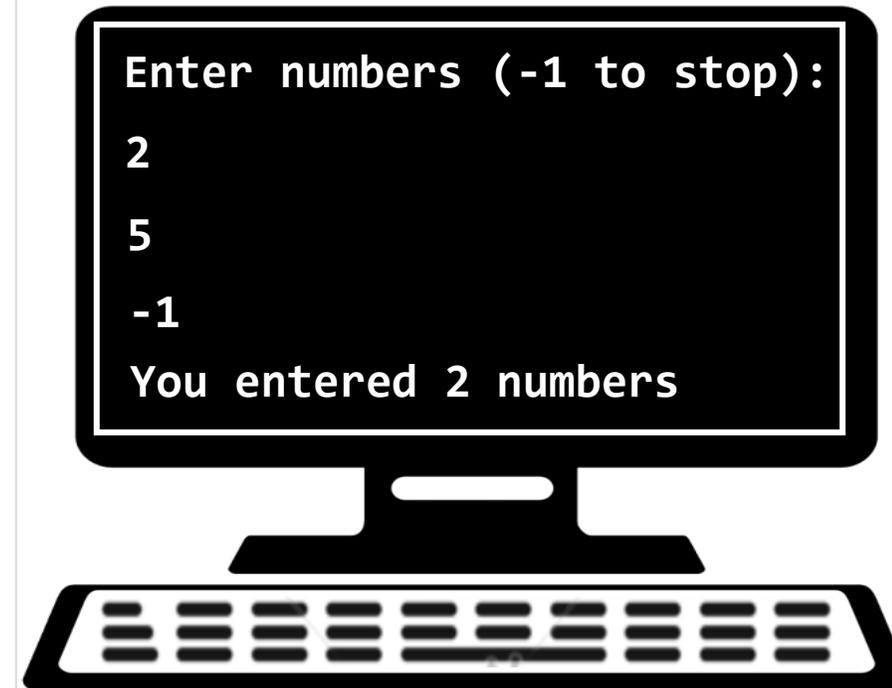
```
Enter numbers (-1 to stop):

2

5

-1

You entered 2 numbers
```

Q. Write a C++ program that inputs a student's name and three quiz marks, calculates and displays the average, and repeats the process while the user chooses to continue.

```cpp
string name;
int quiz1, quiz2, quiz3;
double average;
char again; // To hold Y/N

do {
    cout << "Input student name: ";
    cin >> name;

    cout << "Enter the mark of 3 quizzes: \n";
    cin >> quiz1 >> quiz2 >> quiz3;

    // Calculate and display the average
    average = (quiz1 + quiz2 + quiz3) / 3.0;
    cout << "Name: " << name << ".\t The average: " << average << ".\n";

    // Ask if the user wants to calculate another average
    cout << "Do you want to average another set? (Y/N) ";
    cin >> again;

} while (again == 'Y' || again == 'y');
```

```
Input student name: Haval
Enter the mark of 3 quizzes:
92
71
84
Name: Haval       The average: 82.3333
Do you want to average another set? (Y/N) y
Input student name: Layla
Enter the mark of 3 quizzes:
65
72
94
Name: Layla       The average: 77
Do you want to average another set? (Y/N) n
```

Q. Write a C++ program that lets the user enter integers until -1 is entered, then displays how many numbers were entered (excluding -1).

```cpp
int number;
int count = 0;

cout << "Enter numbers (-1 to stop):\n";

do {
    cin >> number;

    if (number != -1) {
        count = count + 1;
    }
} while (number != -1);

cout << "You entered " << count << " numbers.\n";
```

```
Enter numbers (-1 to stop):
3
8
4
5
9
-1
You entered 5 numbers.
```

Q. Convert the code below to a while loop.

```cpp
int grade, count = 0;
int total = 0;
double avg;

do {
    cout << "Input the grade (-1 to stop): ";
    cin >> grade;

    if (grade != -1) {
        total = total + grade;
        count++;
    }

} while (grade != -1);

if (count > 0) {
    avg = double(total) / count;
    cout << "You input " << count
         << " numbers and the AVG= " << avg;
}
```

Activities and Next Lecture's Topic

Activities

- **Review this lecture note**

- **Practice**

Next Lecture's Topic

- **Iterative Control Structures**
  (nested for loop)

References

- **Gaddis, T. (2014). Starting out with C++: Early objects (7th ed.). Pearson Education.**

- **GeeksforGeeks. (2025, October 8). while Loop in C. GeeksforGeeks. https://www.geeksforgeeks.org/c/c-while-loop/**