

**EEE: 103**

Computer Programming

# **L4: Control Statements**

**Part 1: Decision Making**

Prepared by: Sk Tahmed Salim Rafid

# What are Control Statements?

---

- 🕒 ▶ Control statements are used to control the flow of execution in a program
- 🕒 ▶ They allow the program to make decisions based on certain conditions
- 🕒 ▶ Without control statements, programs execute sequentially line by line
- 🕒 ▶ Control statements enable programs to be dynamic and responsive

# Relational Operators

Used to compare two values and return true (1) or false (0)

<

**Less than**

$5 < 10 \rightarrow \text{true}$

>

**Greater than**

$15 > 10 \rightarrow \text{true}$

<=

**Less than or equal to**

$5 <= 5 \rightarrow \text{true}$

>=

**Greater than or equal to**

$10 >= 5 \rightarrow \text{true}$

==

**Equal to**

$5 == 5 \rightarrow \text{true}$

!=

**Not equal to**

$5 != 10 \rightarrow \text{true}$

# Types of Decision-Making Control Statements

---

**if**

Execute code only if condition is true

**if...else**

Choose between two alternatives

**if...else if...else**

Handle multiple conditions

**switch**

Select from multiple discrete options

# The if Statement

The if statement allows a program to execute a block of code only if a specific condition is true.

## Syntax:

```
if (condition) {  
    // statements to execute if condition is true  
}
```

## Explanation:

- The condition inside the parentheses is evaluated
- If the condition is true, the statements inside { } are executed
- If the condition is false, the program skips the if block
- Program continues with the next statement after the if

# if Statement - Example

---

*Check if a number is positive*

```
1 #include <stdio.h>
2
3 void main() {
4     int number;
5     printf("Enter a number: ");
6     scanf("%d", &number);
7
8     if (number > 0) {
9         printf("The number is positive.\n");
10    }
11 }
```

## Output:

Input: 5  
The number is positive.

Input: -3

# if Statement - Flow



→ User enters a number



→ Program checks: number > 0



→ If TRUE → prints "The number is positive."



→ If FALSE → skips the print statement



→ Program continues to end

# The if...else Statement

The if...else statement allows the program to choose between two alternatives.

## Syntax:

```
if (condition) {  
    // executed if condition is true  
} else {  
    // executed if condition is false  
}
```

## Explanation:

- First, the condition is evaluated
- If the condition is true, the if block executes and else is skipped
- If the condition is false, the else block executes
- Only ONE block executes, never both

# if...else Statement - Example

*Check if a number is even or odd*

```
1 #include <stdio.h>
2
3 void main() {
4     int number;
5     printf("Enter a number: ");
6     scanf("%d", &number);
7
8     if (number % 2 == 0) {
9         printf("The number is even.\n");
10    } else {
11        printf("The number is odd.\n");
12    }
13 }
```

## Output:

Input: 4

The number is even.

Input: 7

The number is odd.

## **if...else Statement - Flow**



→ Program checks if number is divisible by 2



→ If TRUE → prints "The number is even."



→ If FALSE → prints "The number is odd."



→ Only one block executes, never both



→ This is called binary decision making

# The if...else if...else Statement

When there are more than two conditions, we use if...else if...else.

## Syntax:

```
if (condition1) {  
    // executed if condition1 is true  
} else if (condition2) {  
    // executed if condition2 is true  
} else if (condition3) {  
    // executed if condition3 is true  
} else {  
    // executed if all conditions are false  
}
```

## Explanation:

- Conditions are checked in order from top to bottom
- The first true condition executes its block
- Once a condition is true, remaining conditions are skipped
- If no condition is true, the else block executes

## if...else if...else - Example

Grade calculation based on marks

```
1 #include <stdio.h>
2
3 void main() {
4     int marks;
5     printf("Enter your marks: ");
6     scanf("%d", &marks);
7
8     if (marks >= 80) {
9         printf("Grade: A\n");
10    } else if (marks >= 60) {
11        printf("Grade: B\n");
12    } else if (marks >= 50) {
13        printf("Grade: C\n");
14    } else {
15        printf("Grade: F\n");
16    }
17 }
```

### Output:

Input: 65

Grade: B

## if...else if...else - Flow



→ Step 1: Checks marks  $\geq 80$



→ → If TRUE: prints "Grade A" and skips remaining blocks



→ → If FALSE: moves to next condition



→ Step 2: Checks marks  $\geq 60$



→ Step 3: Checks marks  $\geq 50$



→ Step 4: If all conditions fail → executes else → "Grade F"



→ Important: Only ONE block executes!

# The switch Statement

The switch statement is used to select one action from multiple discrete options based on the value of a variable.

## Syntax:

```
switch (expression) {  
    case value1:  
        // code for value1  
        break;  
    case value2:  
        // code for value2  
        break;  
    default:  
        // code if no case matches  
}
```

## Explanation:

- The expression is evaluated once
- Program compares it to each case value
- If a match is found → executes that case block
- break ensures the program exits after the matched case
- default executes if no case matches

# switch Statement - Example

*Day of the week finder*

```
1 #include <stdio.h>
2
3 void main() {
4     int day;
5     printf("Enter day number (1-7): ");
6     scanf("%d", &day);
7
8     switch(day) {
9         case 1:
10            printf("Sunday\n");
11            break;
12        case 2:
13            printf("Monday\n");
14            break;
15        case 3:
16            printf("Tuesday\n");
17            break;
18        case 4:
19            printf("Wednesday\n");
20            break;
21        case 5:
22            printf("Thursday\n");
23            break;
24        case 6:
25            printf("Friday\n");
26            break;
27        case 7:
28            printf("Saturday\n");
29            break;
30        default:
31            printf("Invalid day number\n");
32    }
33 }
```

## Output:

Input: 1

Sunday

Input: 5

Thursday

Input: 9

Invalid day number

# Important Notes on switch

## ! **break is crucial**

Without break, execution "falls through" to the next case

## ! **Expression type**

Switch works with int, char - not with float or string

## ! **Case values**

Must be constant values, not variables or expressions

## ! **default is optional**

But recommended for handling unexpected values

## ! **Multiple cases**

Can have multiple cases execute same code by omitting break

# Summary: Control Statements

Statement	How it works	When to use
<code>if</code>	Executes block if condition is true	One condition to check
<code>if...else</code>	Executes if block if true, else block if false	Two options
<code>if...else if...else</code>	Checks conditions in order, executes first true block	Multiple options
<code>switch</code>	Executes case block matching the expression	Discrete values, many options

# Practice Problems

1

Question 1: Write a program to check if a number is divisible (by 5) (by 5 or 7) (by 5 and 7 both)

3

Question 3: Write a program to check if a year is a leap year (The year is multiple of 400 or the year is a multiple of 4 and not a multiple of 100.)

5

Question 5: Write a program to calculate electricity bill based on units consumed

7

Question 7: Write a program to check if a character is vowel or consonant (ASCII)

2

Question 2: Write a program to find the largest of two numbers

4

Question 4: Write a program to display the month name using switch (1-12)

6

Question 6: Write a program to create a simple calculator using switch (+, -, \*, /) [Use char for operation]

8

Question 8: Write a program to display menu options and perform actions using switch (BKASH/\*121#)

# Thank You

Keep Coding!