# Lecture 4

## The if-else if Ladder in C &

## Switch

In C programming, when you have a series of mutually exclusive conditions, you use the else if structure.

This ensures that the computer only executes the first block that meets the condition and ignores the rest.

**How the Logic Flows:**

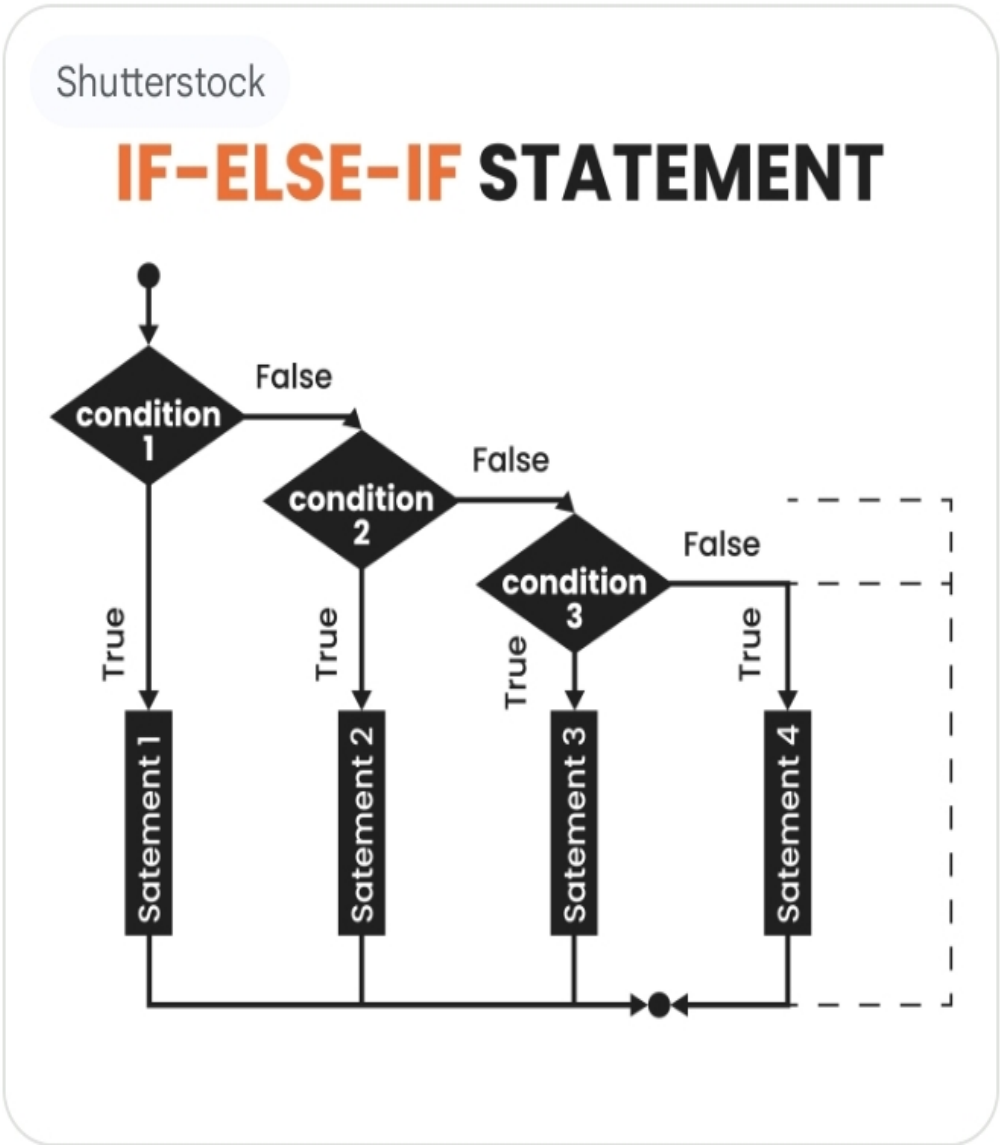
Step 1: The program evaluates the first if statement.

If it is true, it executes the code inside { } and jumps to the end of the entire block.

Step 2: If the first condition is false, it moves to the next else if.

Step 3: This process continues line by line.

Final Outcome: If all conditions are false and there is no final else statement, the program will execute nothing and simply move on to the next part of your code.



## Syntax and Code Example

Here is a practical example in C.

Let's say we are checking the temperature:

```
#include <stdio.h>
int main() {
    int temp = 25;
    if (temp > 40) {
        printf("It is very hot.\n");  // Checks first
    }
    else if (temp > 20) {
        printf("The weather is nice.\n"); // Executed ONLY if first is false
    }
    else if (temp > 0) {
        printf("It is cold.\n"); // Executed ONLY if previous are false
    }
```

```
    // If none are true, nothing is printed.
    return 0;
}
```

**Key Rules in C:**

Sequential: The order matters! The program always starts from the top.

The "Exit" behavior: Once a condition is met, the rest of the ladder is not even tested, which saves processing time.

No "Else" requirement: As you requested, if you don't add a final else { ... } block, the program will do absolutely nothing if all conditions fail.

**Example**

```
#include <stdio.h>
int main() {
    int color;
      printf("Enter color number (1-3): ");
       scanf("%d", &color);
    if (color == 1) {
        printf("Action: Stop (Red)\n");
    }
    else if (color == 2) {
        printf("Action: Wait (Yellow)\n");
    }
    else if (color == 3) {
        printf("Action: Go (Green)\n");
    }
    else {
```

```
        printf("Invalid input! Please choose 1, 2, or 3.\n");
    }


    return 0;
}
```

## Interactive Grade Calculator with "Out of Range"

This version explicitly prints "Out of range" if the user enters a score that doesn't fit into the A-F categories, specifically by using the final else statement to handle scores below 60.

## Code Example

```
#include <stdio.h>
int main() {
    int score;
1 //    . Get input from the user
    printf("Enter your numerical score (0-100):
");
    scanf("%d", &score);

2 //    . The Conditional Ladder (If-Else If)
    if (score >= 90 && score <= 100) {
        printf("Grade: A\n");
    }
    else if (score >= 80 && score < 90) {
        printf("Grade: B\n");
    }
    else if (score >= 70 && score < 80) {
```

```
        printf("Grade: C\n");
    }
    else if (score >= 60 && score < 70) {
        printf("Grade: D\n");
    }
    else {
        // This is the final else, handling any
score below 60 or invalid input
        printf("Out of range\n");
    }

    return 0;
}
```

**Breakdown of the Structure**:

Sequential Check: The code checks score 90 =< , then score >= 80, and so on.

Final else: If the score is 59 or less (e.g., 50), the program skips all previous conditions and executes the else block, printing "Out of range.

## Switch Statement

The switch statement is a better way of writing a program which employs an if-else ladder.

It is C's built-in multiple branch decision statement.

The syntax for the switch statement is as follows:

```
switch (integer expression) {

case constant1:

statement1;

break;

case constant2:

statement2;

break;


...
```

default:

statement;

}

The keyword break should be included at the end of each case statement.

In general, whenever a break statement is encountered in C, it interrupts the

normal flow of control.

In the switch statement, it causes an exit from the switch shunt.

The default clause is optional.

The right brace at the end marks the end of switch statement.

Example

```
#include <stdio.h>
int main() {
    int day;
    printf("Enter a number (1-7) to get the day of the week: ");
    scanf("%d", &day);
  switch (day) {
        case 1:
            printf("Saturday\n");
            break;
        case 2:
            printf("Sunday\n");
            break;
        case 3:
            printf("Monday\n");
            break;
```

```
        case 4:
            printf("Tuesday\n");
            break;
        case 5:
            printf("Wednesday\n");
            break;
        case 6:
            printf("Thursday\n");
            break;
        case 7:
            printf("Friday\n");
            break;
        default:
            printf("Invalid input! Please enter a
number between 1 and 7.\n");
    }


    return 0;
```

}

## switch Statement Operation

The switch statement works as follows:

1) An integer control expression is evaluated.

2) A match is looked for between this expression value and the case

constants.

If a match is found, execute the statements for that case.

If a match is not found, execute the default statement.

3) Terminate switch when a break statement is encountered or by “falling out

the end”.

Some things to be aware of when using a switch statement:

- case values must be unique (How to decide otherwise?)

- switch statement only tests for equality

- The control expression can be of type character since they are  internally treated as integers.

Example

```
#include <stdio.h>
int main() {
    char firstLetter;


1 //    . Get the first letter of a color from the user
    printf("Enter the first letter of a color (o, r, g, b): ");
    scanf(" %c", &firstLetter); // The space handles the newline character
2 //    . Switch statement to match the letter to a color
    switch (firstLetter) {
        case 'o':
            printf("orange\n");
            break;
```

```
  case 'r':
        printf("red\n");
          break;
      case 'g':
        printf("Green\n");
          break;
      case 'b':
         printf("blue\n");
          break;
      default:
          printf("Color not found for that
letter.\n");
   }


   return 0;
}
```

## Homework Assignment: Simple Menu Selector

Task:

Write a C program that displays a simple menu for a coffee shop.

The user should enter a number (1-3) to select an item.

Menu:

* Espresso

* Cappuccino

* Latte

Requirements:

Use a switch statement to print the name of the selected item based on the number entered.

If the user enters a number outside the range (1-3), print: "Invalid choice."