Week 4: Control Statements (Looping & Branching)

Overview

Welcome to Week 5 of C programming! This week, we will explore control statements that help in decision-making and iteration. By the end of this tutorial, you will:

- Understand how to use loops (`for`, `while`, `do-while`) to repeat tasks
- Learn how to use conditional statements (`if-else`, `switch`) for decision-making
- Utilize the conditional (`?:`) operator for compact conditional expressions

Time Breakdown

- **Introduction to Loops & Conditionals (10 min)**
- **Using `for` Loops (15 min)**
- **Nested Loops & Multiplication Tables (15 min)**
- **Working with `if-else` and `switch` (20 min)**
- **Exercises & Q/A (20 min)**

---

1. Understanding the For Loop (15 min)

A `for` loop is used when the number of iterations is known beforehand.

Example: Print even numbers from 2 to 20

```c
#include <stdio.h>

int main() {
    for (int i = 2; i <= 20; i += 2) {
        printf("%d ", i);
    }
    return 0;
}
```

PROF

Expected Output:

```
2 4 6 8 10 12 14 16 18 20
```

Exercise 1: Modify the above program to calculate and print the sum of all even numbers from 2 to 20.

---

2. Nested Loops & Multiplication Table (15 min)

Nested loops allow us to create tables and patterns.

Example: Multiplication table up to 10x10

```c
#include <stdio.h>

int main() {
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= 10; j++) {
            printf("%3d ", i * j);
        }
        printf("\n");
    }
    return 0;
}
```

Expected Output:

```
  1   2   3   4   5   6   7   8   9  10
  2   4   6   8  10  12  14  16  18  20
...
```

Exercise 2: Modify the above program to allow the user to specify the table size.

---

3. Decision Making with If-Else (20 min)

`if-else` statements help in executing different code blocks based on conditions.

Example: Check if a number is positive, negative, or zero

PROF

```c
#include <stdio.h>

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (num > 0) {
        printf("The number is positive.\n");
    } else if (num < 0) {
        printf("The number is negative.\n");
    } else {
        printf("The number is zero.\n");
    }
```

```
    return 0;
}
```

Example Output:

```
Enter a number: -5
The number is negative.
```

Exercise 3: Write a program that checks if an exam score is a pass or fail (passing mark: 50).

---

4. Switch Statement (15 min)

The `switch` statement is useful when there are multiple possible conditions.

Example: Identify vowels and consonants

```c
#include <stdio.h>

int main() {
    char ch;
    printf("Enter a character: ");
    scanf(" %c", &ch);

    switch(ch) {
        case 'a': case 'e': case 'i': case 'o': case 'u':
        case 'A': case 'E': case 'I': case 'O': case 'U':
            printf("It is a vowel.\n");
            break;
        default:
            printf("It is a consonant.\n");
    }
    return 0;
}
```

Example Output:

```
Enter a character: a
It is a vowel.
```

Exercise 4: Write a program that acts as a basic calculator using `switch`.

---

5. Conditional Operator (10 min)

The conditional (`?:`) operator provides a compact way to write `if-else` statements.

Example: Check if a number is positive, negative, or zero

```
#include <stdio.h>

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    printf("%s\n", num > 0 ? "Positive" : num < 0 ? "Negative" :
"Zero");
    return 0;
}
```

Example Output:

```
Enter a number: 7
Positive
```

Exercise 5: Write a program that finds the largest of two numbers using the conditional operator.

Bonus Challenge: Print a Right-Angled Triangle Pattern

Example:

PROF

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter number of rows: ");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Example Output (for `n = 5`):

```
*
**
***
****
*****
```

Summary & Wrap-Up (10 min)

This week, we covered:

- Using loops to automate repetitive tasks
- Making decisions with `if-else` and `switch`
- Writing concise conditions with the `?:` operator

Next Week: Functions and Modular Programming

Happy coding! 🚀