C++ 101 – Session 2 Notes

1. Recap of Previous Session

In our last session, we covered:

- The basic structure of a C++ program
- Variable declaration and assignment
- Common data types (`int`, `float`, `char`, `string`, `bool`)
- Comments (single-line and multi-line)
- Expressions: Arithmetic, Comparison, Logical
- Conditional statements using `if`, `if-else`, and `else-if`

---

2. The `switch` Statement

The `switch` statement allows you to test a variable against multiple constant values.

📌 **Syntax**:

```
switch (expression) {
    case value1:
        // code block
        break;
    case value2:
        // code block
        break;
    ...
    default:
        // code block
}
```

✅ Example: Days of the Week

```
int day = 3;

switch (day) {
    case 1:
        cout << "Monday";
        break;
    case 2:
        cout << "Tuesday";
        break;
    case 3:
        cout << "Wednesday";
        break;
    case 4:
        cout << "Thursday";
        break;
    case 5:
        cout << "Friday";
        break;
    case 6:
        cout << "Saturday";
        break;
    case 7:
        cout << "Sunday";
        break;
    default:
        cout << "Invalid day";
}
```

💡 Notes:

- `break` prevents the program from executing the next case.
- `default` handles any value not matched by a `case`. `default` runs if no case matches.

---

3. The `while` Loop

A `while` loop repeats a block of code **as long as a condition is true**.

📌 Syntax:

```
while (condition) {

    // code block

}
```

✅ Example:

```
int i = 0;

while (i < 5) {

    cout << i << endl;

    i++;

}
```

💡 Notes:

The condition is checked **before** each loop iteration.

If the condition is false from the start, the loop does not run at all.

🧠 Tip:

Use `while` when:

You don’t know how many times you'll loop in advance.

You want to loop based on user input or a condition changing during runtime.