Tutorial 9 - C++ Control Structures, If-Else, and Switch-Case

Control Structures in C++

Control structures define the flow and logic of a program.
Three types of control structures:

1. **Sequence Structure**:
    - Instructions are executed sequentially, one after another.
    - Example:

      ```
      int a = 10;
      cout << "Value: " << a;  // Executes in order.
      ```

    - **Diagram**: Straight flow without branching or loops.
2. **Selection Structure**:
    - Executes instructions based on conditions (true/false).
    - Implemented using:
        - **If-Else Statements**
        - **Switch-Case Statements**
    - **Diagram**: Flow branches based on conditions.
3. **Loop Structure**:
    - Repeats instructions while a condition is true.
    - Example:

      ```
      while (x < 5) {
          cout << x++;
      }
      ```

    - **Diagram**: Circular flow until the condition becomes false.

If-Else Statements

- Used to implement **selection structures**.
- Syntax:

  ```
  if (condition) {
      // Code block if condition is true
  } else if (another_condition) {
      // Code block if another condition is true
  } else {
      // Code block if no conditions are true
  }
  ```

- Example Program:

  ```
  int age;
  cout << "Enter your age: ";
  cin >> age;
  ```

```
if (age < 18) {
    cout << "You cannot come to the party.\n";
} else if (age == 18) {
    cout << "You are a kid and will get a kid pass.\n";
} else {
    cout << "You can come to the party.\n";
}

```

- Key Points:
  - Conditions are evaluated in order.
  - Supports multiple `else if` branches.
  - Final `else` is executed if no conditions are true.

Switch-Case Statements

- Used to test a variable's value against multiple cases.
- Syntax:

```
switch (variable) {
    case value1:
        // Code for case 1
        break;
    case value2:
        // Code for case 2
        break;
    default:
        // Code if no cases match
        break;
}

```

- Example Program:

```
int age;
cout << "Enter your age: ";
cin >> age;

switch (age) {
    case 18:
        cout << "You are 18\n";
        break;
    case 22:
        cout << "You are 22\n";
        break;
    default:
        cout << "No special cases\n";
        break;
}
cout << "Done with switch case\n";

```

- Key Points:
  - `break` prevents execution of other cases.
  - `default` handles unmatched cases.
  - Suitable for discrete, predefined values.

Code Example

```cpp
#include <iostream>
using namespace std;

int main() {
    int age;
    cout << "Tell me your age: ";
    cin >> age;

    // 1. If-Else Statement
    if (age < 18) {
        cout << "You cannot come to the party.\n";
    } else if (age == 18) {
        cout << "You are a kid and will get a kid pass.\n";
    } else {
        cout << "You can come to the party.\n";
    }

    // 2. Switch-Case Statement
    switch (age) {
        case 18:
            cout << "You are 18\n";
            break;
        case 22:
            cout << "You are 22\n";
            break;
        case 2:
            cout << "You are 2\n";
            break;
        default:
            cout << "No special cases\n";
            break;
    }

    cout << "Done with switch case.";
    return 0;
}

```

Short Notes

Control Structures

1. **Sequence**: Executes one instruction after another.
2. **Selection**: Executes based on a condition (e.g., If-Else, Switch-Case).
3. **Loop**: Repeats instructions while a condition is true.

If-Else

- Syntax:

  ```cpp
  if (condition) { ... }
  else if (condition) { ... }
  else { ... }

  ```

- Allows multiple conditions with `else if`.
- Example: `if (age < 18) { ... }`

Switch-Case

- Tests a variable against specific cases.
- Syntax:

```
switch (variable) {
    case value1: ... break;
    case value2: ... break;
    default: ... break;
}

```

- Use `break` to exit cases.
- Example: `switch (age) { case 18: ... }`