

Lecture – 12 –

Multiple Conditions

- Sometimes there is a need to test a variable for equality against multiple values.
- That can be achieved using multiple if statements.
- **Example:**

```
int age = 42;
if (age == 16) {
    cout << "Too young";
}
if (age == 42) {
    cout << "Adult";
}
if (age == 70) {
    cout << "Senior";
}
```

- The **switch** statement is a more elegant solution in this scenario.

The **switch** Statement

- The **switch** statement tests a variable against a list of values, which are called **cases**, to determine whether it is equal to any of them.

```
switch (expression) {
    case value1:
        statement(s);
        break;
    case value2:
        statement(s);
        break;
    ...
    case valueN:
        statement(s);
        break;
}
```

Switch evaluates the expression to determine whether it's equal to the value in the case statement.

- If a match is found, it executes the statements in that case.
- A switch can contain any number of **case** statements, which are followed by the **value** in question and a **colon**.
- Here is the previous example written using a single **switch** statement:

```

int age = 42;
switch (age) {
    case 16:
cout << "Too young";
        break;
    case 42:
        cout << "Adult";
        break;
    case 70:
        cout << "Senior";
        break;
}

```

- The code above is equivalent to three **if** statements.
- Notice the keyword **break**; that follows each case. That will be covered shortly.

The default Case

- In a switch statement, the optional **default** case can be used to perform a task when none of the cases is determined to be true.
- **Example:**

```

int age = 25;
switch (age) {
    case 16:
cout << "Too young";
        break;
    case 42:
        cout << "Adult";
        break;
    case 70:

```

```

        cout << "Senior";
        break;
    default:
        cout << "This is the default case";
    }

```

// Outputs "This is the default case"

- The **default** statement's code executes when none of the cases matches the switch expression.
- The **default** case must appear at the end of the switch.

The **break** Statement

- The **break** statement's role is to terminate the switch statement.
- In instances in which the variable is equal to a case, the statements that come after the case continue to execute until they encounter a **break** statement.
- In other words, leaving out a **break** statement results in the execution of all of the statements in the following cases, even those that don't match the expression.
- **Example:**

```

        int age = 42;
        switch (age) {
            case 16:
                cout << "Too young" << endl;
            case 42:
                cout << "Adult" << endl;
            case 70:
                cout << "Senior" << endl;
            default:
                cout << "This is the default case" << endl;
        }
        /* Outputs
           Adult
           Senior
           This is the default case
        */

```

- As you can see, the program executed the matching case statement, printing "Adult" to the screen.
- With no specified **break** statement, the statements continued to run after the matching case.
- Thus, all the other case statements printed.
- This type of behavior is called **fall-through**.
- As the switch statement's final case, the **default** case requires no **break** statement.
- The **break** statement can also be used to break out of a loop.