Control Flow - Decision Making

What is Decision Making in Programming?

In real life, we take decisions every day.

Examples:

- If it is raining, take an umbrella
- If marks are above 35, pass the exam

Programs also need to make such decisions.

Decision making allows a program to choose what to do based on a condition.

Why Decision Making is Needed

A program should not run everything blindly.

It must:

- Check conditions
- Choose the correct path
- Skip unnecessary code

This makes programs:

- Smarter
- More useful
- More realistic

The if Statement

The if statement checks a condition.

If the condition is **true**, the code inside runs.

Syntax:

```
if (condition) {
   // code runs if condition is true
}
```

Simple example:

```
int age = 20;
if (age >= 18) {
   System.out.println("Eligible to vote");
```

```
}
```

Explanation:

- age >= 18 → condition
- If true → message is printed

(If nothing prints, it simply means the condition was false.)

---

if–else Statement

Used when there are **two possible outcomes**.

Syntax:

```
if (condition) {
   // runs if true
} else {
   // runs if false
}
```

Example:

```
int marks = 30;
if (marks >= 35) {
   System.out.println("Pass");
} else {
   System.out.println("Fail");
}
```

Here:

- Either if runs
- Or else runs

Never both.

Nested if Statement

Sometimes, one decision depends on another.

This is called **nested if**.

Example:

```
int marks = 85;
if (marks >= 35) {
   if (marks >= 75) {
```

```
        System.out.println("Distinction");
    }
}
```

Explanation:

- First checks pass or fail
- Then checks distinction

(If this feels long, don't worry — nesting becomes natural with practice.)

The switch Statement

switch is used when:

- One variable
- Multiple fixed values

It is cleaner than many if-else statements.

Syntax:

```
switch(value) {
    case 1:
        // code
        break;
}
```

Example:

```
int day = 2;
switch(day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    default:
        System.out.println("Invalid day");
}
```

Explanation:

- day is checked

- Matching case runs
- break stops execution

(Forgetting break is a common beginner mistake.)

---

if vs switch (Quick Comparison)

if–else	**switch**
Works with ranges	Works with fixed values
More flexible	Cleaner for menus
Used for conditions	Used for choices

Boolean Conditions (Important)

Decision making always depends on **true or false**.

Example:

```
boolean isLoggedIn = true;

if (isLoggedIn) {
   System.out.println("Welcome");
}
```

If condition is false, code inside does not run.

Remember This

Decision making means:
Check condition → choose path → run code

Once this is clear, all control flow becomes easy.