

Control Statements

Control Statements →

instructions that alter the normal sequential flow of execution. They allow you to make decisions, repeat code blocks, or jump to specific sections of your program.

For eg. 1) If it's raining, wear a raincoat.
Else, wear a T-shirt.

2) If the username & password match, grant access. Else, display an error message.

3) If the product is in stock, allow purchase. Else, display an "Out of stock" message.

Control statements available in Java

- Decision making - Used to make decisions based on conditions
 - if statement
 - if-else statement
 - Switch statement

- Looping Statements - Used to repeatedly execute a block of code.
 - for loop.
 - while loop.
 - do-while loop.
 - For-each loop.

- Jump statements - Used to alter the normal flow of execution.

- break
 - continue
 - return
- } statement.

if statement

"conditional statement" that executes a block of code if a specific condition is true."

Ex. Checking if coffee shop customer is eligible for a discount.

```
// Amount spent by customer
int purchaseAmount = 100;
```

```
// whether customer is a student
boolean isStudent = true;
```

```
if (purchaseAmount >= 50 && isStudent) {
    cout << "you are eligible for a 10% discount!";
} else {
    cout << "Sorry, you are not eligible for discount";
}
```


Ex. Grade calculation based on marks.

```
int marks = 85;
```

```
if (marks >= 90) {  
    println("Grade: A");  
}
```

```
else if (marks >= 80) {  
    println("Grade: B");  
}
```

```
else if (marks >= 70) {  
    println("Grade: C");  
}
```

```
else if (marks >= 60) {  
    println("Grade: D");  
}
```

```
else {  
    println("Grade: F");  
}
```