Resource Software Solution

Session 02: Decision Making

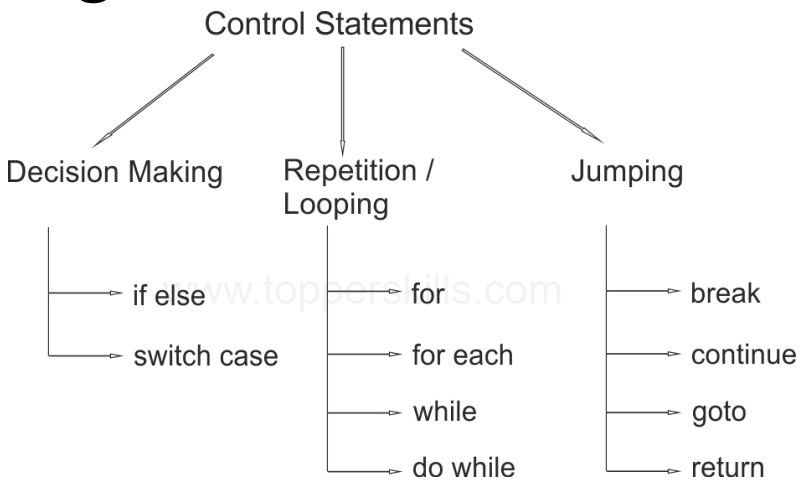
Objectives

1. Overview
2. If-else statement
3. Switch-case statement

Overview (1)

Flow Control Statements

- All application development environments provide a decision making process called **flow control statements** that direct the application execution
- Flow control enables a developer to create an application that can examine[kiểm tra] the existing conditions, and decide a suitable course of action



Overview (2)

Decision Making

01 if

02 if-else

03 Nested-if

04 if-else-if

05 switch-case

Resource Software Solution

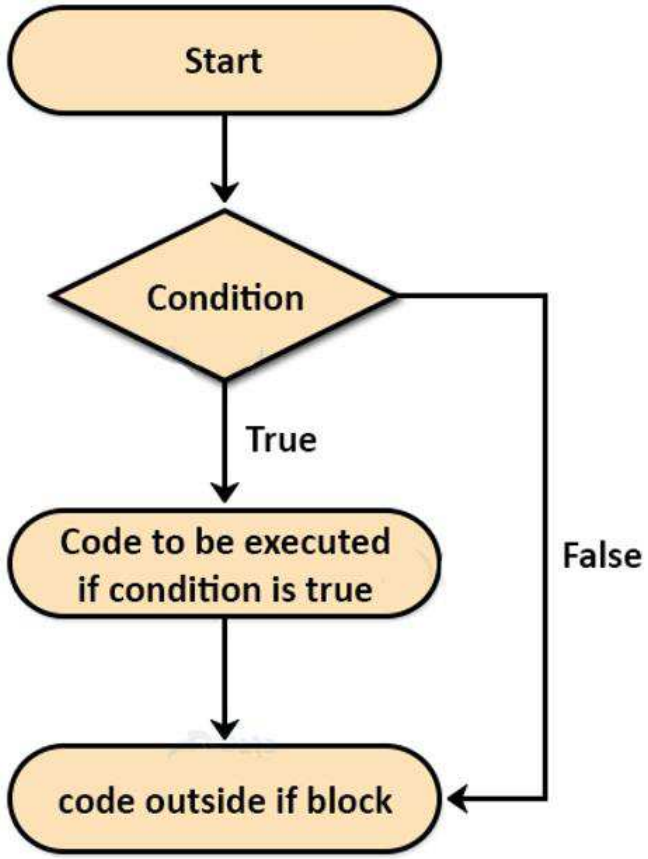
2

If-else statement

If statement in Java (1)

- It encompasses a boolean condition followed by a scope of code which is **executed only** when the **condition evaluates to true**
- Syntax

```
if (condition) {
  // code to be executed
}
```



If statement in Java (2)

- Example Java program to understand if statement

```
public class IfStatement {
  public static void main(String[] args) {
    String s = "R2S Academy";

    if (s.equals("R2S Academy")) {
      System.out.println("The string is R2S Academy");
    }
  }
}
```

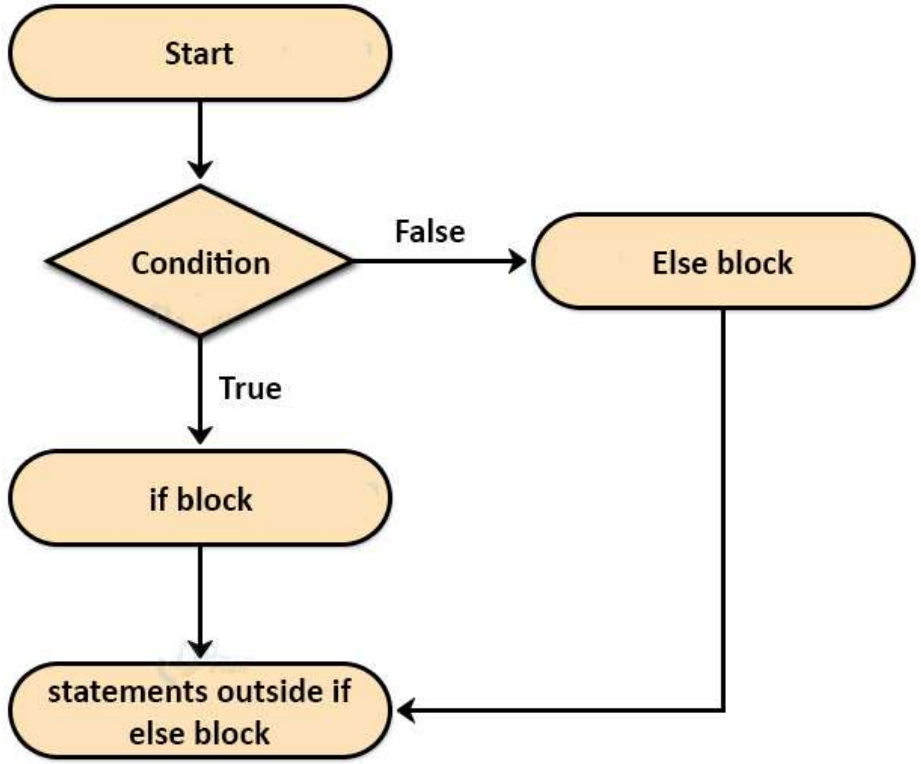
Output:

The string is R2S Academy

If else statement in Java (1)

- One being the code to be executed if the **condition evaluates to true** and the **other one to be executed** if the **value is false**
- Syntax

```
if (condition) {
  // code to be executed if the condition is true
} else {
  // code to be executed if the condition is false
}
```



If else statement in Java (2)

- Example Java program to understand if-else statement

```
public class IfStatement {
  public static void main(String[] args) {
    String s = "R2S Company";

    if (s.equals("R2S Academy")) {
      System.out.println("The string is R2S Academy");
    } else {
      System.out.println("The string is not R2S Academy");
    }
  }
}
```

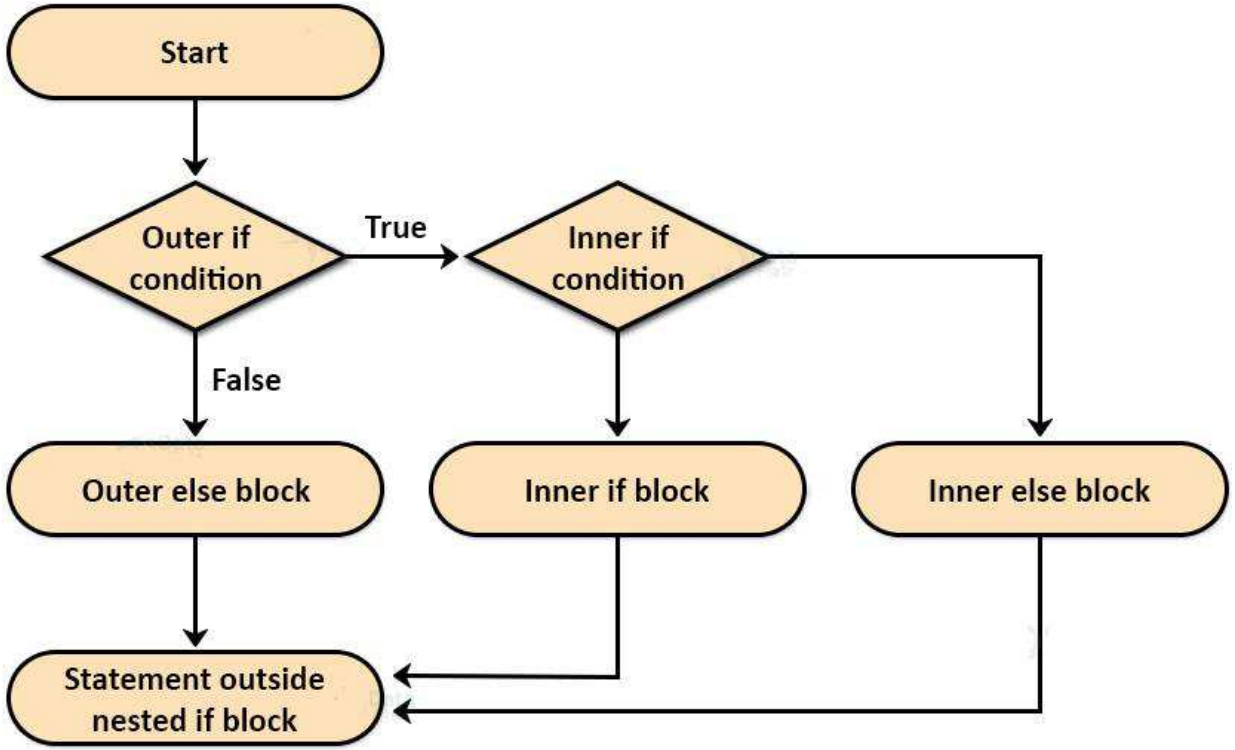
Output:

The string is not R2S Academy

Nested if statement in Java (1)

- If the condition of the **outer** if statement evaluates to **true** then the **inner if statement is evaluated.** Nested if's are important if we have to declare extended conditions to a previous condition
- Syntax

```
if (condition) {
  // code to be executed
  if (condition) {
    // code to be executed
  }
}
```



Nested if statement in Java (2)

- Example Java program to understand nested if statement

```
public class IfStatement {
  public static void main(String[] args) {
    String s = "R2S Company";

    if (s.equals("R2S Academy")) {
      System.out.println("The string is R2S Academy");
      if (s.charAt(0).equals("R") {
        System.out.println("The first character is R!");
      }
    }
  }
}
```

Output:

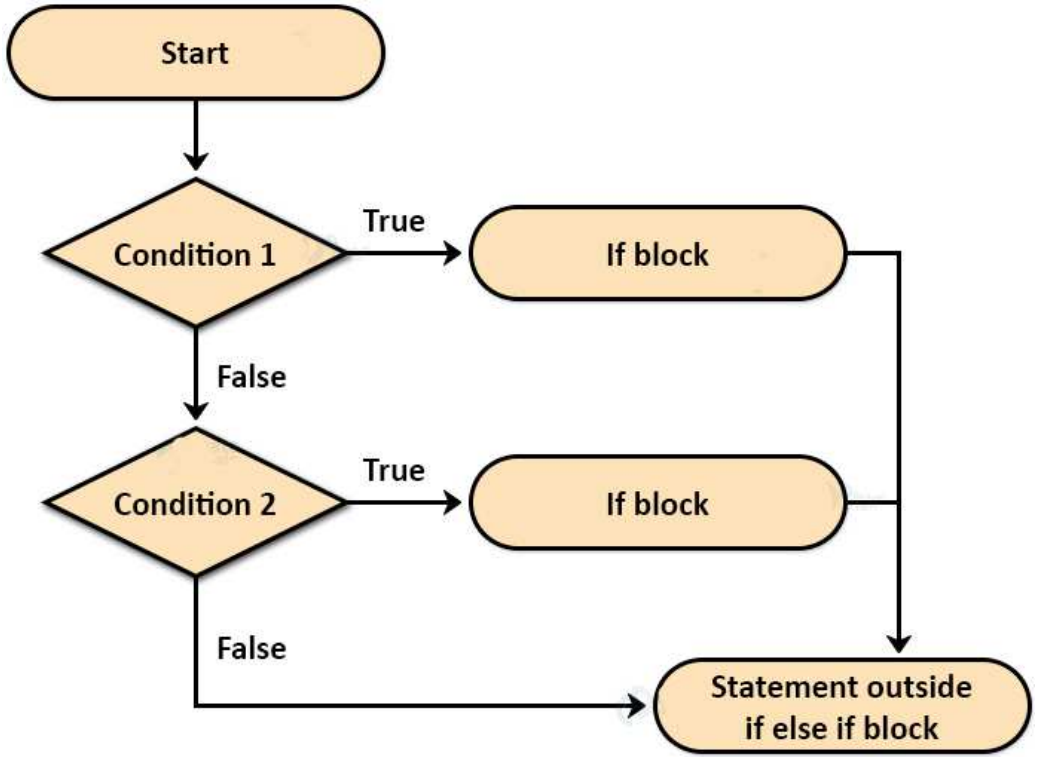
The string is R2S Academy

The first character is R!

If-else-if statement in Java (1)

- These statements are similar to the if else statements. The only difference lies in the fact that each of the else statements can be paired with a different if condition statement
- Syntax

```
if (condition) {
  // code to be executed
} else if (condition) {
  // code to be executed
} else if (condition) {
  // code to be executed
} else {
  // code to be executed
}
```



if-else-if statement in Java (2)

- Example Java program to understand if-else-if statement

```
public class IfStatement {
  public static void main(String[] args) {
    int n1 = 1, n2 = 2, n3 = 3;

    if (n1 > n2 && n1 > n3) {
      System.out.println("Max is " + n1);
    } else if (n2 > n1 && n2 > n3) {
      System.out.println("Max is " + n2);
    } else if (n3 > n1 && n3 > n2) {
      System.out.println("Max is " + n3);
    }
  }
}
```

Output:

Max is 3

Resource Software Solution

3

Switch statement in Java

Switch statement in Java (1)

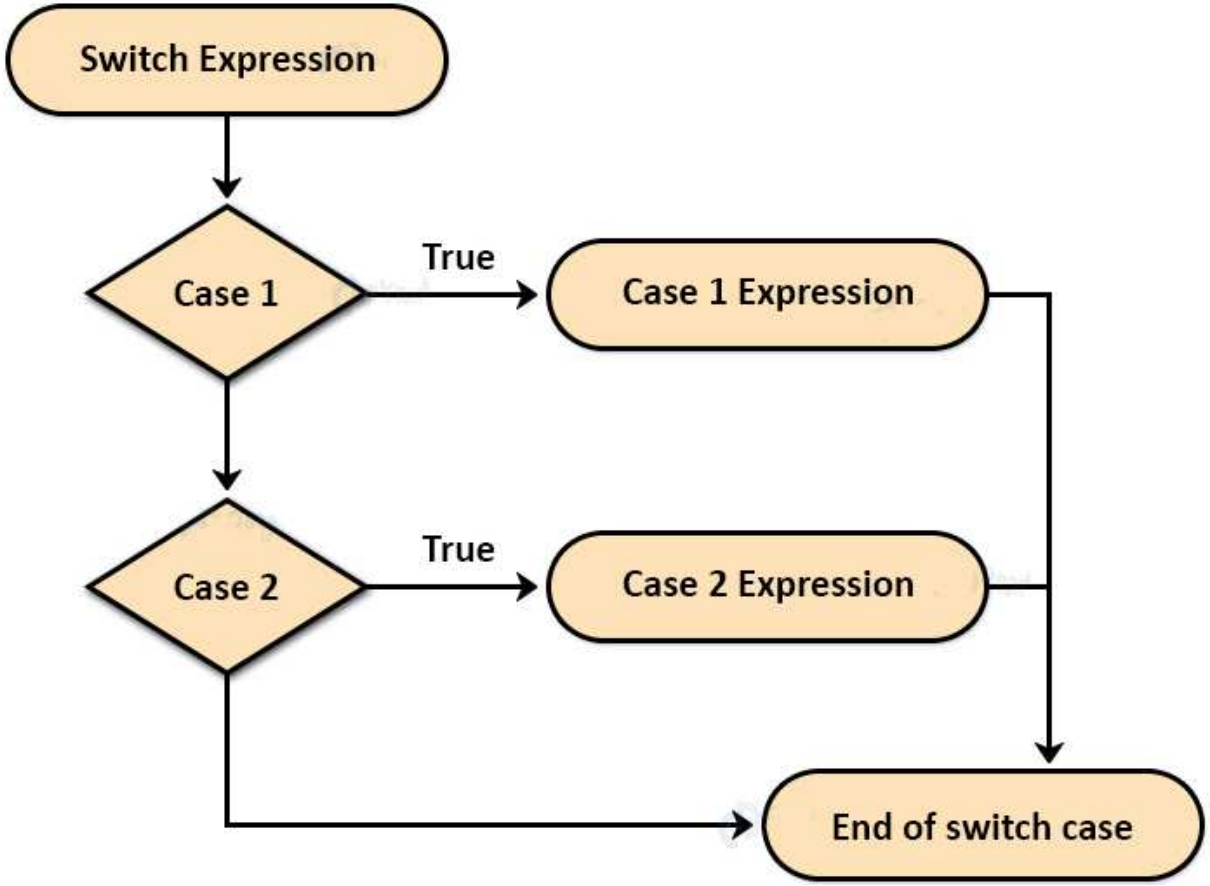
Introduction

- If the output matches with any of the cases mentioned, then the particular block is executed. A break statement is written after every end of the case block so that the remaining statements are not executed
- The default case is written which is executed if none of the cases are the result of the expression. This is generally the block where error statements are written

Switch statement in Java (2)

Syntax & Flow Diagram

```
switch(expression) {
 case <value1>:
   // code to be executed
   break;
 case <value2>:
  // code to be executed
  break;
 default:
  // code to be defaultly executed
}
```



Switch statement in Java (3)

Example

```
public class SwitchStatement {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System. in );
    System.out.println("1. Java");
    System.out.println("2. C#");
    System.out.print("Your choice:");
    int ch = sc.nextInt();
```

```
    switch (ch) {
      case 1:
        System.out.println("Java!");
        break;
      case 2:
        System.out.println("C#!");
        break;
       default:
         System.out.println("Wrong input!");
    }
  }
}
```

Output:

1. Java

2. Python

Your choice: 1

Java!

Thankyou!