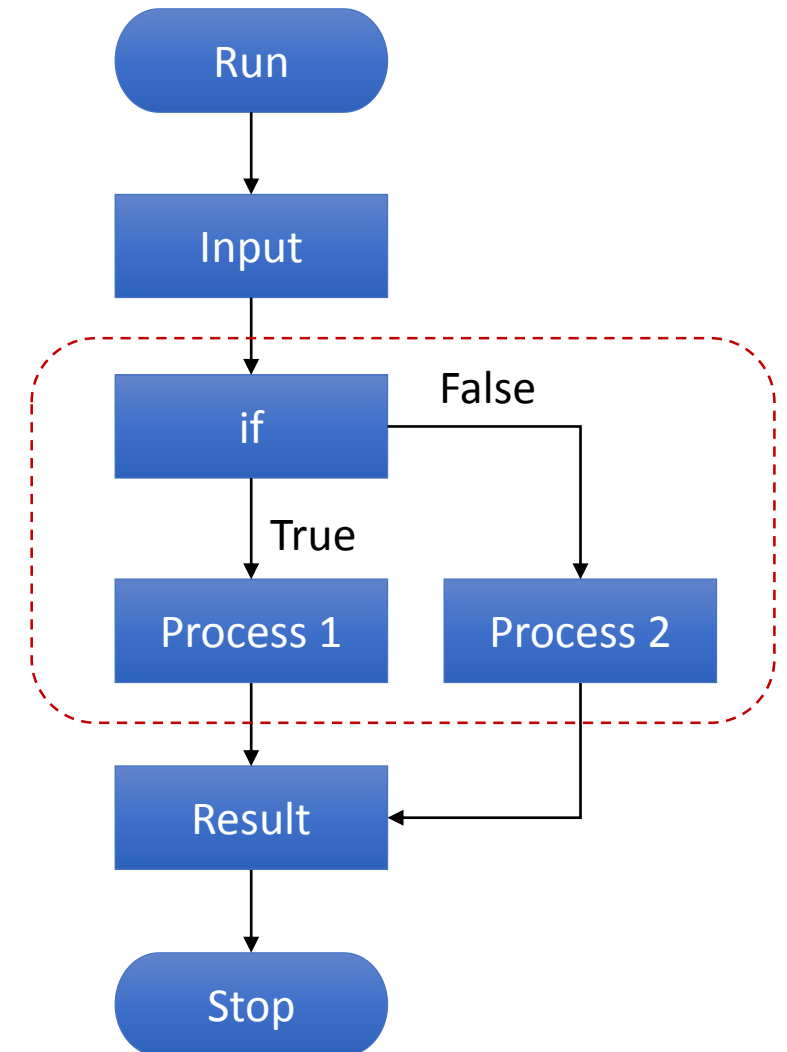


# if Statement

Sungchul Lee

# Learning Object

- Single-Selection Statement (if)
  - ❑ Control Flow
- Double-Selection Statement (if/else)
  - ❑ Control Flow
- Multi-Selection Statement (if/else if/else)
  - ❑ Control Flow
- Nested if statement
  - ❑ Control Statement in other Control



# Single-Selection Statement (if)

## ➤ Syntax

**if** (boolean expression) {

    //statements will execute if the boolean expression is true

}

➤ An if statement consists of a Boolean expression followed by one or more statements

➤ Example:

```
if (score >= 60) {  
    System.out.println ( "Passed");  
}
```

# Practice1

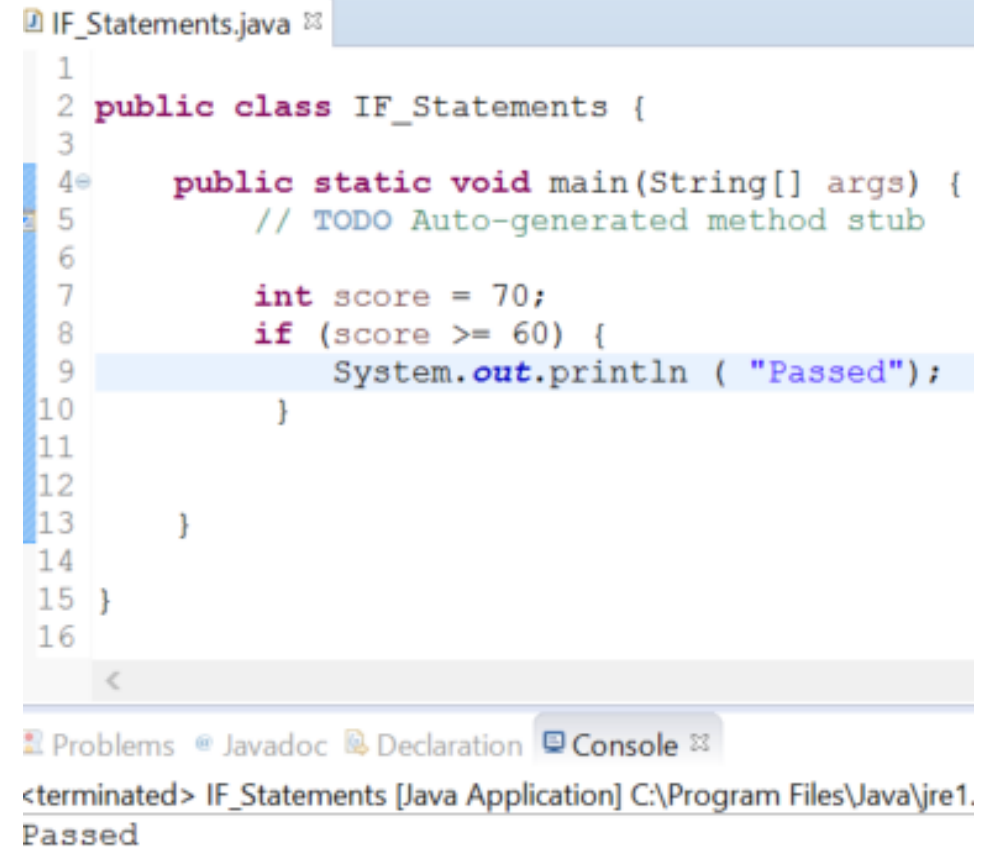
1. Make a new project
  - ❑ Project name: IF\_Statements1
2. Create a new Class File
  - ❑ Class name: IF\_Statements
3. Coding:

```
public class IF_Statements {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        int score = 50;  
        if (score >= 60) {  
            System.out.println ( "Passed");  
        }  
    }  
}
```

# Practice1

➤ When condition is true

```
if (score >= 60) {  
    System.out.println ( "Passed");  
}
```

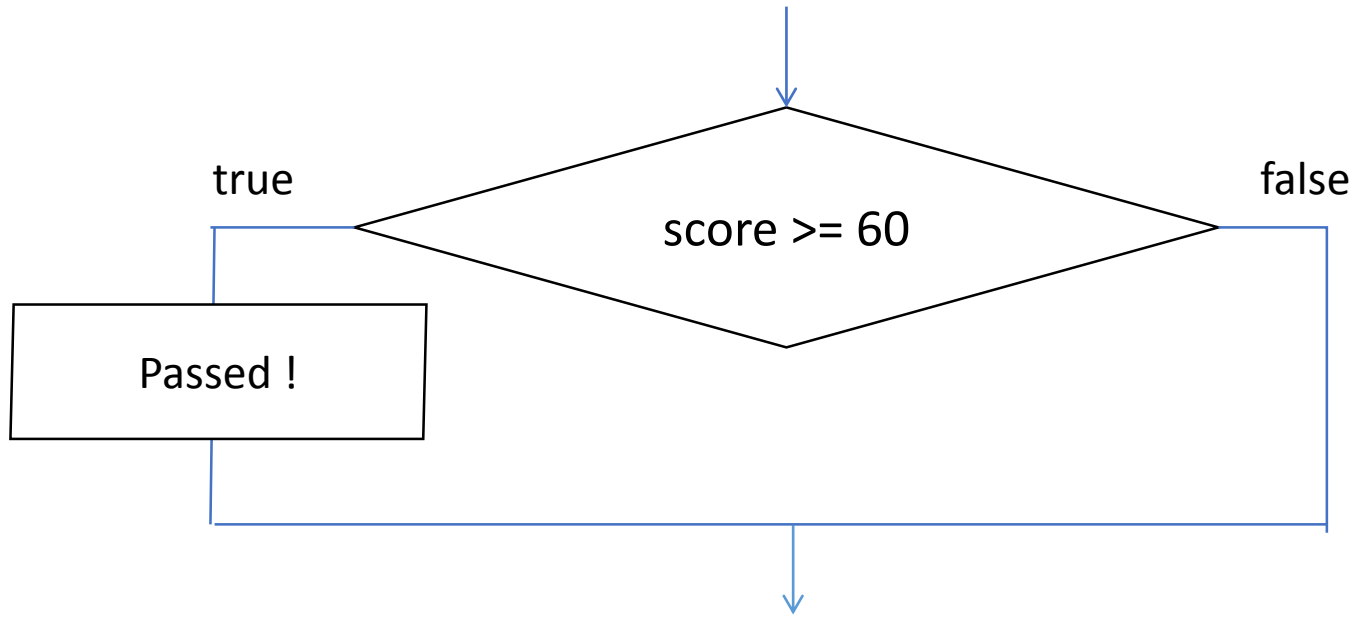


The screenshot shows an IDE window titled 'IF\_Statements.java'. The code is as follows:

```
1  
2 public class IF_Statements {  
3  
4     public static void main(String[] args) {  
5         // TODO Auto-generated method stub  
6  
7         int score = 70;  
8         if (score >= 60) {  
9             System.out.println ( "Passed");  
10        }  
11  
12    }  
13  
14  
15 }  
16
```

At the bottom, the 'Console' tab is active, displaying the output: `<terminated> IF_Statements [Java Application] C:\Program Files\Java\jre1. Passed`

# Control Flow of if statement



```
if (score >= 60) {  
    System.out.println ( "Passed");  
}
```

# Double-Selection (if/else)

## ➤ Syntax:

```
if (Boolean expression) {  
    //executes when the boolean expression is true }  
else {  
    //executes when the boolean expression is false }
```

## ➤ Double - selection statement specify an action to perform when the condition is true and a different action when the condition is false

## ➤ Example:

```
if (score >= 60) {  
    System.out.println ( "Passed");  
} else {  
    System.out.println("Failed");  
}
```

# Practice1 – cont.

## ➤ More conditions

### ❑ If true

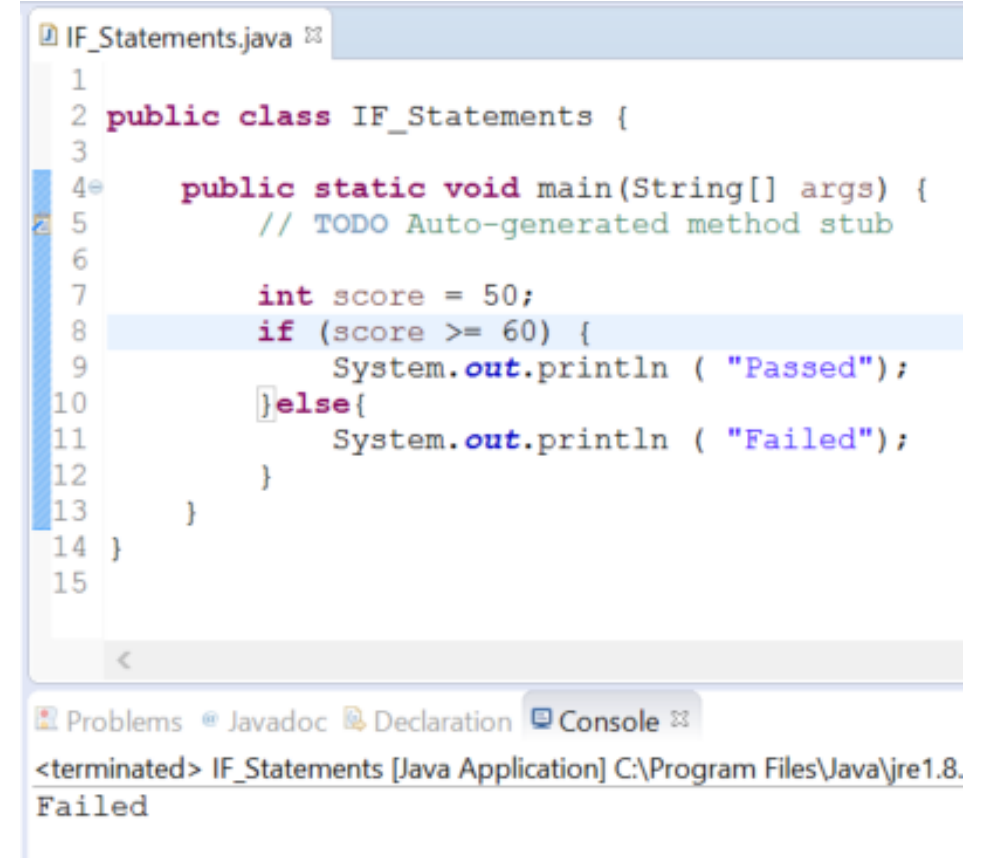
- ❖ Inside of if statement

- ❖ `System.out.println ( "Passed");`

### ❑ If false

- ❖ Inside of else statement

- ❖ `System.out.println ( "Failed");`



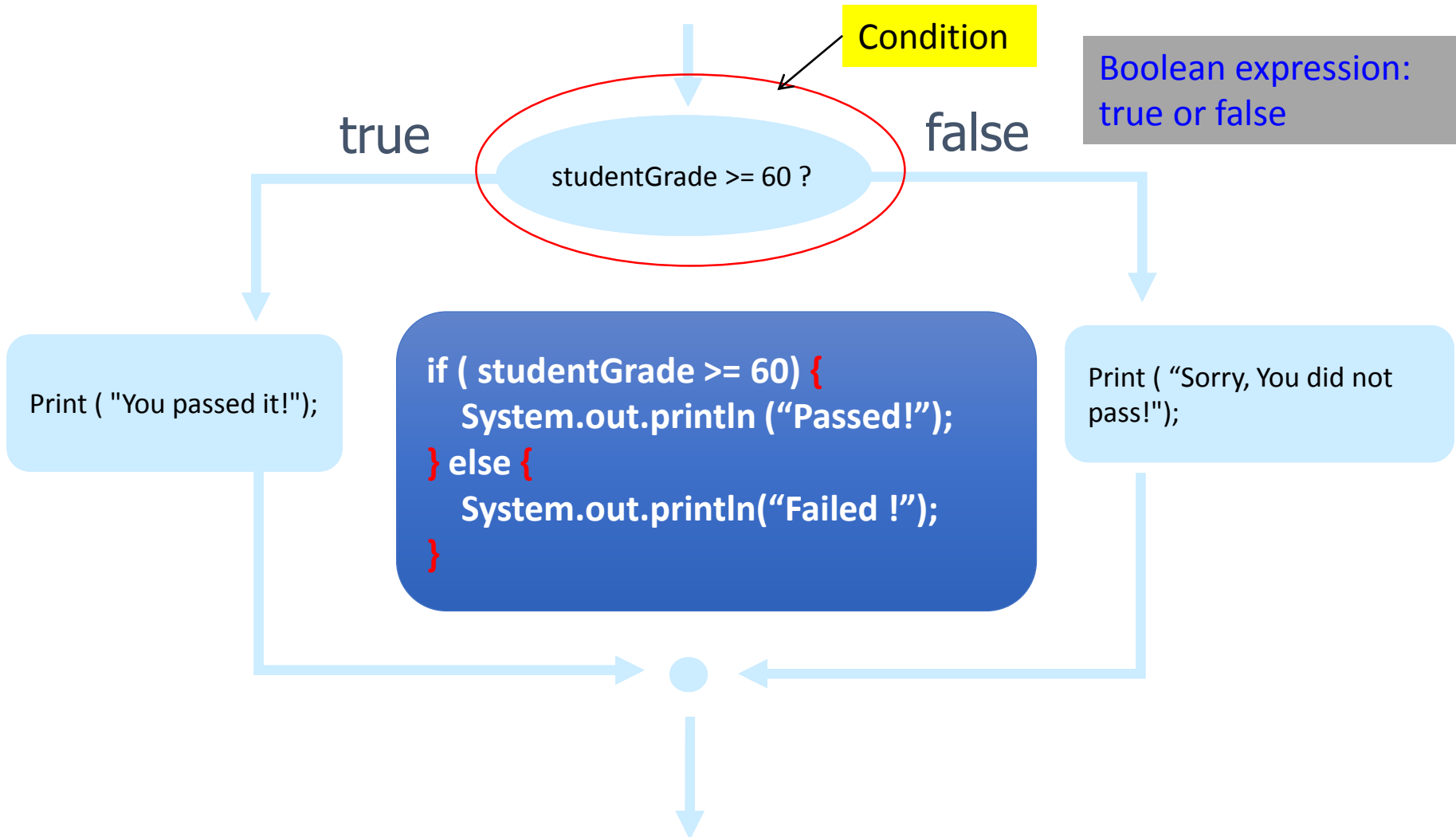
```
IF_Statements.java
1
2 public class IF_Statements {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7         int score = 50;
8         if (score >= 60) {
9             System.out.println ( "Passed");
10        }else{
11            System.out.println ( "Failed");
12        }
13    }
14 }
15
```

Problems Javadoc Declaration Console

<terminated> IF\_Statements [Java Application] C:\Program Files\Java\jre1.8.  
Failed



# Control Flow of if-then-else



# Multi-Selection (if / else if / else)

## ➤ Syntax

```
if (Boolean expression1) {  
    //executes when the boolean expression1 is true  
} else if (Boolean expression2) {  
    //executes when the boolean expression2 is true  
} else if (Boolean expression3) { // make as many as you want  
    //executes when the boolean expression3 is true  
} else {  
    //executes when the boolean expression is false  
}
```

# Multi-Selection (if / else if / else) – cont.

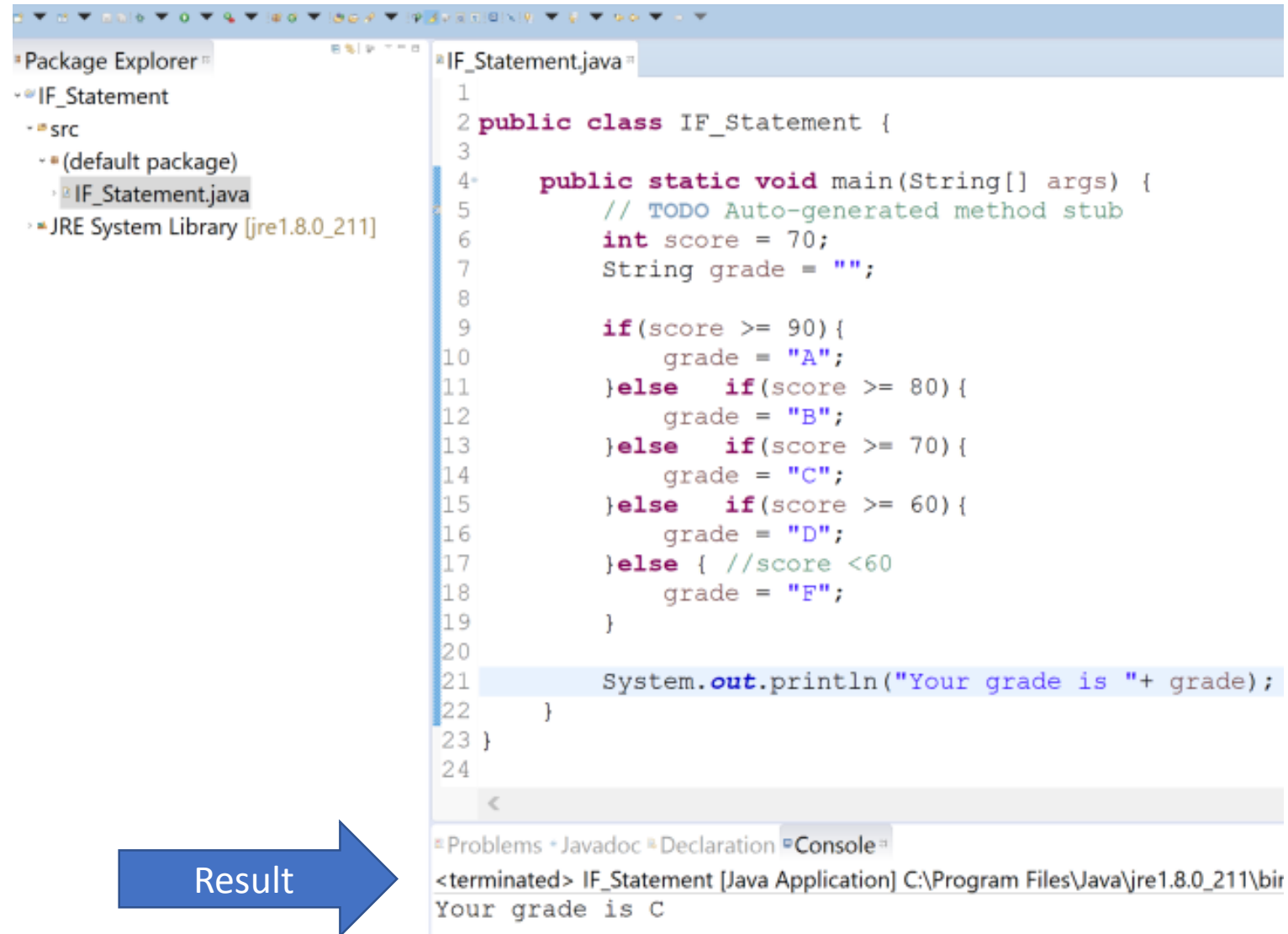
- It is good option, when program needs more than two situations
  - ❑ Need more complicated statements
    - ❖ If/else if/ .../else if/else
- Use Case example of if/else
- Example 1: Ranking of the school
  - ❑ Suppose we have 500 schools
    - ❖ If one school's rank is **100 or lower**, it is a **tier 1** school
    - ❖ If one school's rank is **between 101 and 200**, it is a **tier 2**
    - ❖ If one school's rank is **between 201 and 500**, it is a **tier 3**
- Example 2: Convert a numerical score into a symbolic score (A,B,C,D,F)
  - ❑ **Basic if/else statement is not enough**

# Practice 2

1. Make a new project
  - ❑ Project name: IF\_Statement
2. Create a new Class File
  - ❑ Class name: IF\_Statement
3. Coding:

```
public class Variable {  
    public static void main(String[] args) {  
        int score = 70;  
        String grade = "";  
  
        if(score >= 90){  
            grade = "A";  
        }else if(score >= 80){  
            grade = "B";  
        }else if(score >= 70){  
            grade = "C";  
        }else if(score >= 60){  
            grade = "D";  
        }else { //score <60  
            grade = "F";  
        }  
        System.out.println("Your grade is "+ grade);  
    }  
}
```

# Practice 2 – Code and Result



The screenshot displays an IDE with two main panels. The left panel, 'Package Explorer', shows a project structure with a package 'IF\_Statement' containing a source folder 'src' and a file 'IF\_Statement.java'. The right panel shows the code for 'IF\_Statement.java'.

```
1
2 public class IF_Statement {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int score = 70;
7         String grade = "";
8
9         if(score >= 90){
10             grade = "A";
11         }else if(score >= 80){
12             grade = "B";
13         }else if(score >= 70){
14             grade = "C";
15         }else if(score >= 60){
16             grade = "D";
17         }else { //score <60
18             grade = "F";
19         }
20
21         System.out.println("Your grade is "+ grade);
22     }
23 }
24
```

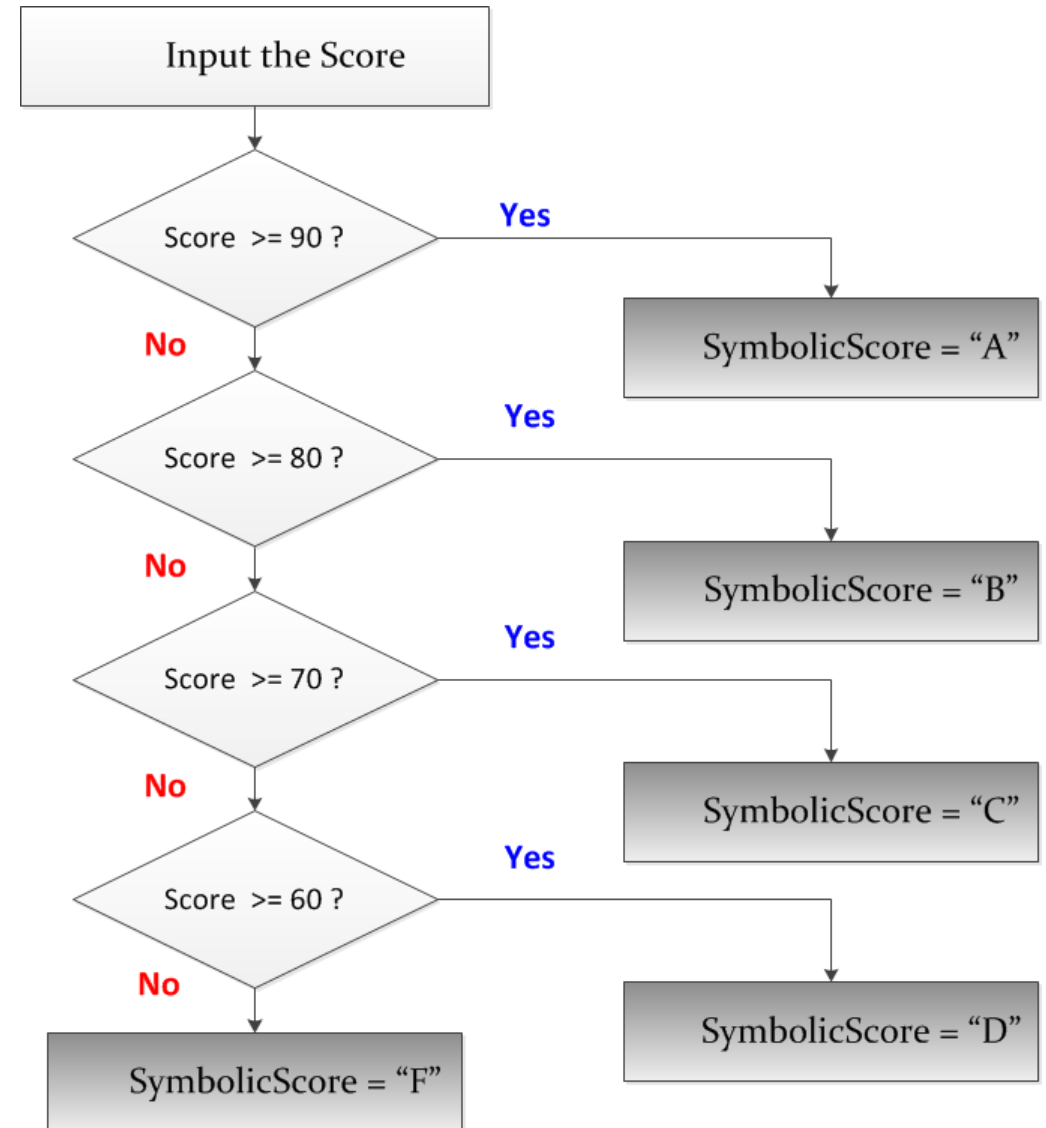
Below the code editor, the 'Console' tab is active, showing the output of the program:

```
<terminated> IF_Statement [Java Application] C:\Program Files\Java\jre1.8.0_211\bin
Your grade is C
```

A blue arrow labeled 'Result' points from the bottom left towards the console output.

# Multi-Selection (if / else if / else) - Example

```
int score = 70;  
String grade = "";  
  
if(score >= 90){  
    SymbolicScore = "A";  
}else if(score >= 80){  
    SymbolicScore = "B";  
}else if(score >= 70){  
    SymbolicScore = "C";  
}else if(score >= 60){  
    SymbolicScore = "D";  
}else { //score <60  
    SymbolicScore = "F";  
}
```



# The Nested-if Statement

## ➤ Syntax

```
if (Boolean expression1) {  
    //executes when the boolean expression1 is true  
    if (Boolean expression2) { // inside if, if-else or else statement  
        //executes when the boolean expression2 is true  
    } else {  
        //executes when the boolean expression is false }  
    } else {  
        //executes when the boolean expression is false }
```

# The Nested-if Statement – cont.

- Control statement can contain **other control statements**.
  - ❑ An if statement containing another if statement is called a nested-if statement.
- Example:

When outer if statement is true

```
if (testScore >= 70) {  
    if (studentAge < 10){ // //test score >= 70 and age < 10  
        System.out.println("You did a great job");  
    } else{//test score >= 70 and age >= 10  
        System.out.println("You did pass");  
    }  
} else { //test score < 70  
    System.out.println("You did not pass");  
}
```



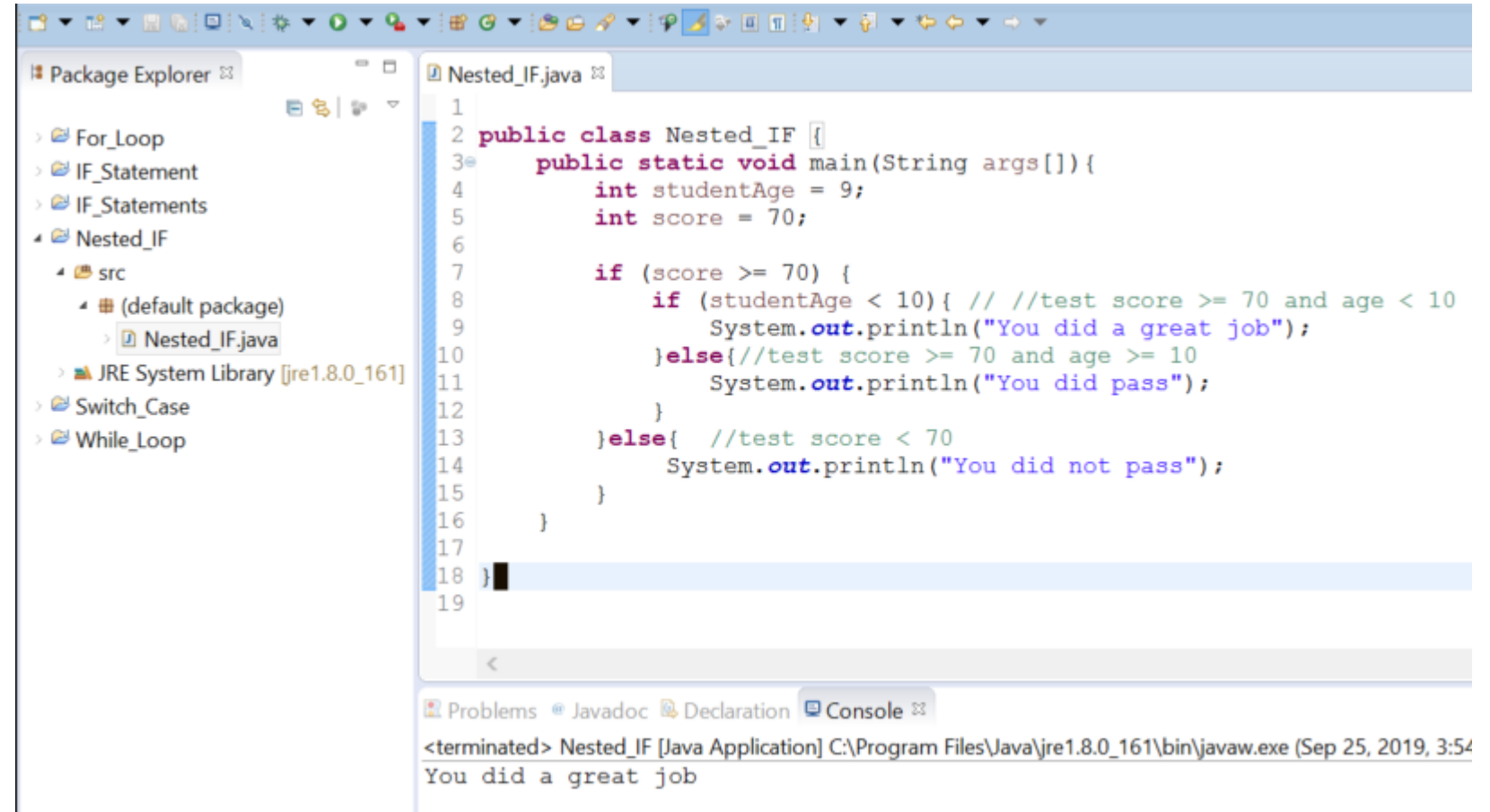
# Practice 3

1. Make a new project  
☐ Project name: **Nested\_IF**
2. Create a new Class File  
☐ Class name: **Nested\_IF**
3. Coding:

```
public class Nested_IF {  
    public static void main(String args[]){  
        int studentAge = 9;  
        int score = 70;  
        if (score >= 70) {  
            if (studentAge < 10){  
                System.out.println("You did a great job");  
            }else{//test score >= 70 and age >= 10  
                System.out.println("You did pass");  
            }  
        }else{ //test score < 70  
            System.out.println("You did not pass");  
        }  
    }  
}
```

# Practice 3

## ➤ Inner –if



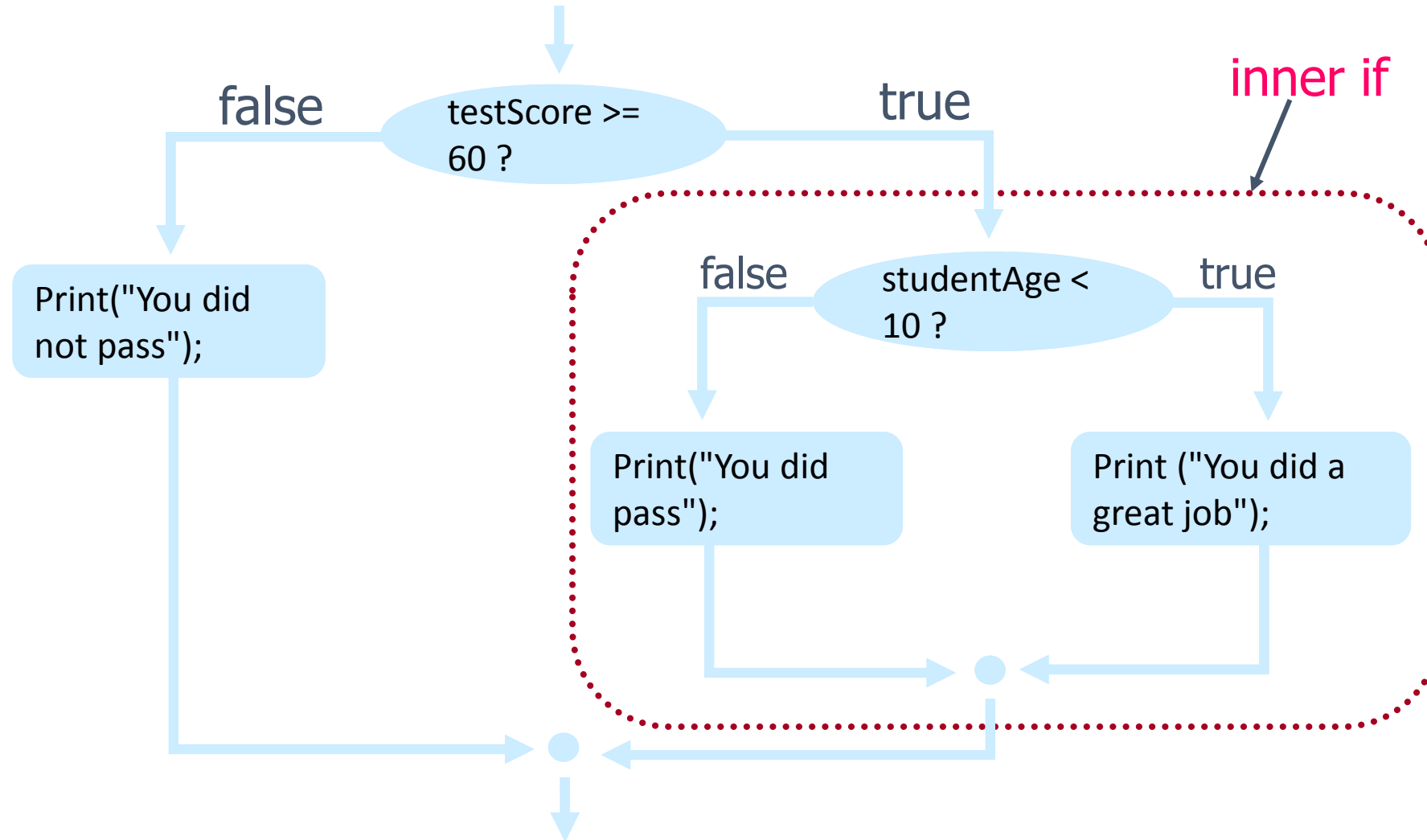
The screenshot shows an IDE with the Package Explorer on the left and the code editor on the right. The Package Explorer shows a project structure with folders for For\_Loop, IF\_Statement, IF\_Statements, and Nested\_IF. The Nested\_IF folder is expanded, showing a src folder with a default package containing Nested\_IF.java. The code editor displays the following Java code:

```
1  
2 public class Nested_IF {  
3     public static void main(String args[]) {  
4         int studentAge = 9;  
5         int score = 70;  
6  
7         if (score >= 70) {  
8             if (studentAge < 10) { // //test score >= 70 and age < 10  
9                 System.out.println("You did a great job");  
10            } else { //test score >= 70 and age >= 10  
11                System.out.println("You did pass");  
12            }  
13        } else { //test score < 70  
14            System.out.println("You did not pass");  
15        }  
16    }  
17  
18 }  
19
```

The Console at the bottom shows the output of the program:

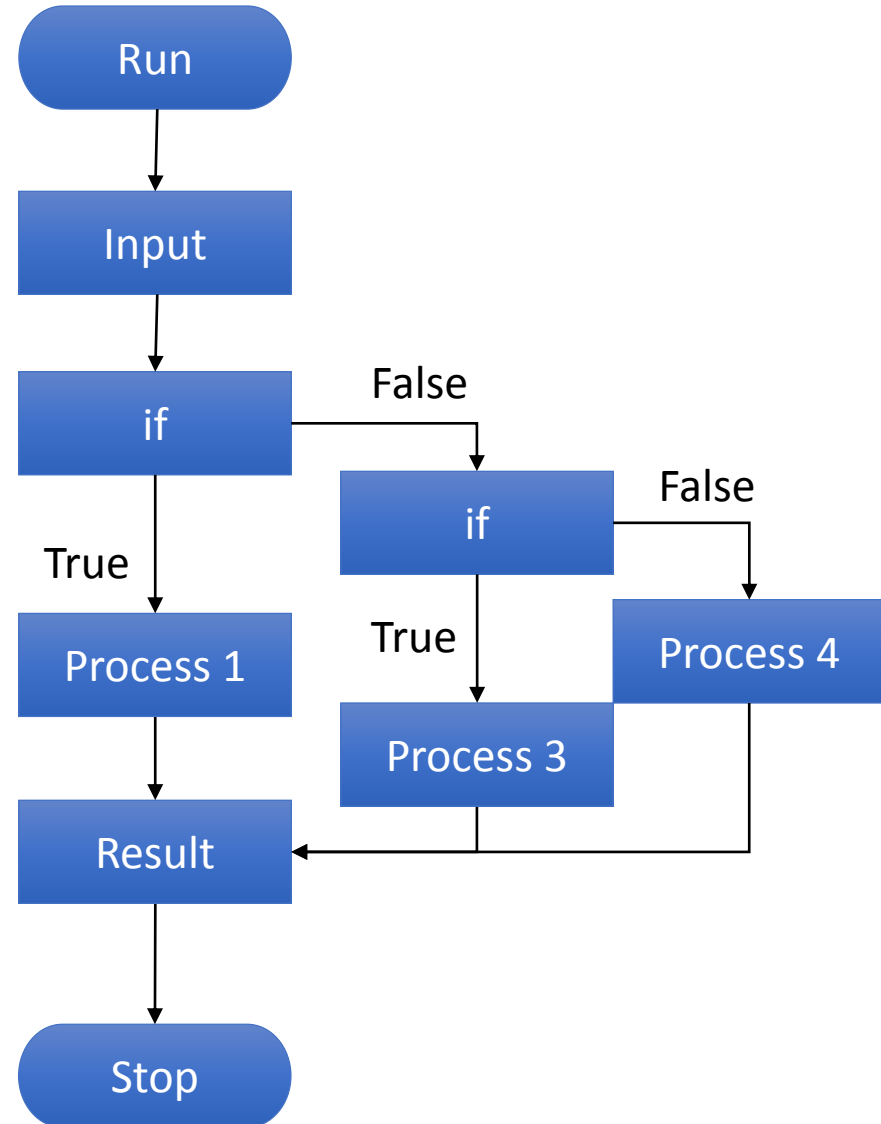
```
<terminated> Nested_IF [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Sep 25, 2019, 3:54  
You did a great job
```

# Flow of the Nested-if



# Flow of the Entire Process with Nested-if

1. Run a program
2. Receive input
3. If Condition
4. True Processing
5. False Processing
6. Add more if condition
7. Show result
8. Stop the program



# Summary

- Single-Selection Statement (if)
- Double-Selection Statement (if/else)
- Multi-Selection Statement (if/else if/else)
- Control Statement in other Control

```
IF_Statement.java
1
2 public class IF_Statement {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int score = 70;
7         String grade = "";
8
9         if(score >= 90){
10             grade = "A";
11         }else if(score >= 80){
12             grade = "B";
13         }else if(score >= 70){
14             grade = "C";
15         }else if(score >= 60){
16             grade = "D";
17         }else { //score <60
18             grade = "F";
19         }
20
21         System.out.println("Your grade is " + grade);
22     }
23 }
```

