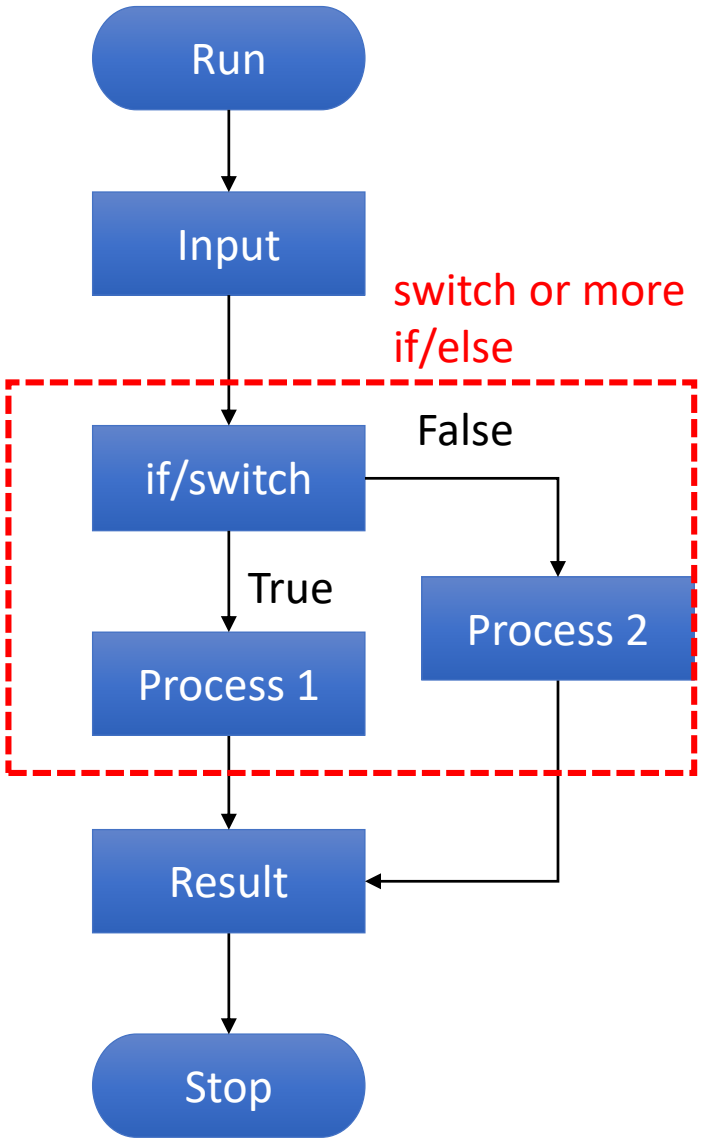
Switch/case Statement

Sungchul Lee

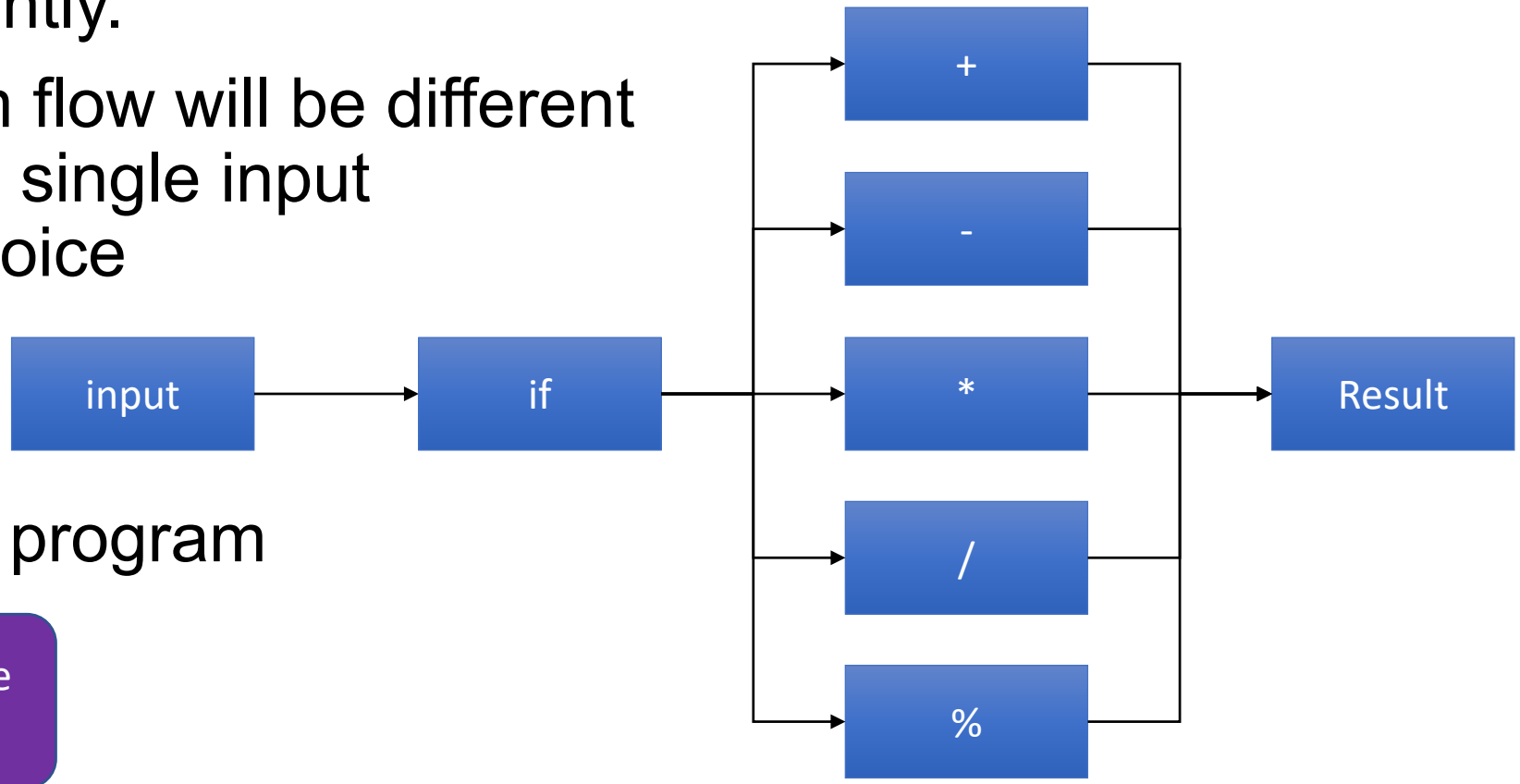
Learning Object

- SWITCH Case Statement
  - Flow
- Break Statement



Multiple Conditions

- Overuse of nested if statements makes a program difficult to read. Java provides a **switch statement** to handle **multiple conditions** efficiently.
- What if a program flow will be different based on a user's single input among various choice



- e.g. calculator program

Programmer can use if/else if/else instead of switch

SWITCH Case Statement

- If a variable has a limited number of values and you want to use “equals” among Relational Operators, you might can use a “switch case” statement
- The **switch statement** observes the following rules:
  - The **switch expression** must yield a value of char, byte, short, int, or String type and must always be enclosed in parentheses.
  - The value1, …, and valueN must have the same type as the value of the switch expression. Note that value1, …., and valueN are constant expressions, meaning that they cannot contain variables in the expression, such as 1 + x. (No arithmetic Operations)
  - The default case, which is optional, can be used to perform actions when none of the specific cases matches the switch expression.

SWITCH Case Statement Syntax

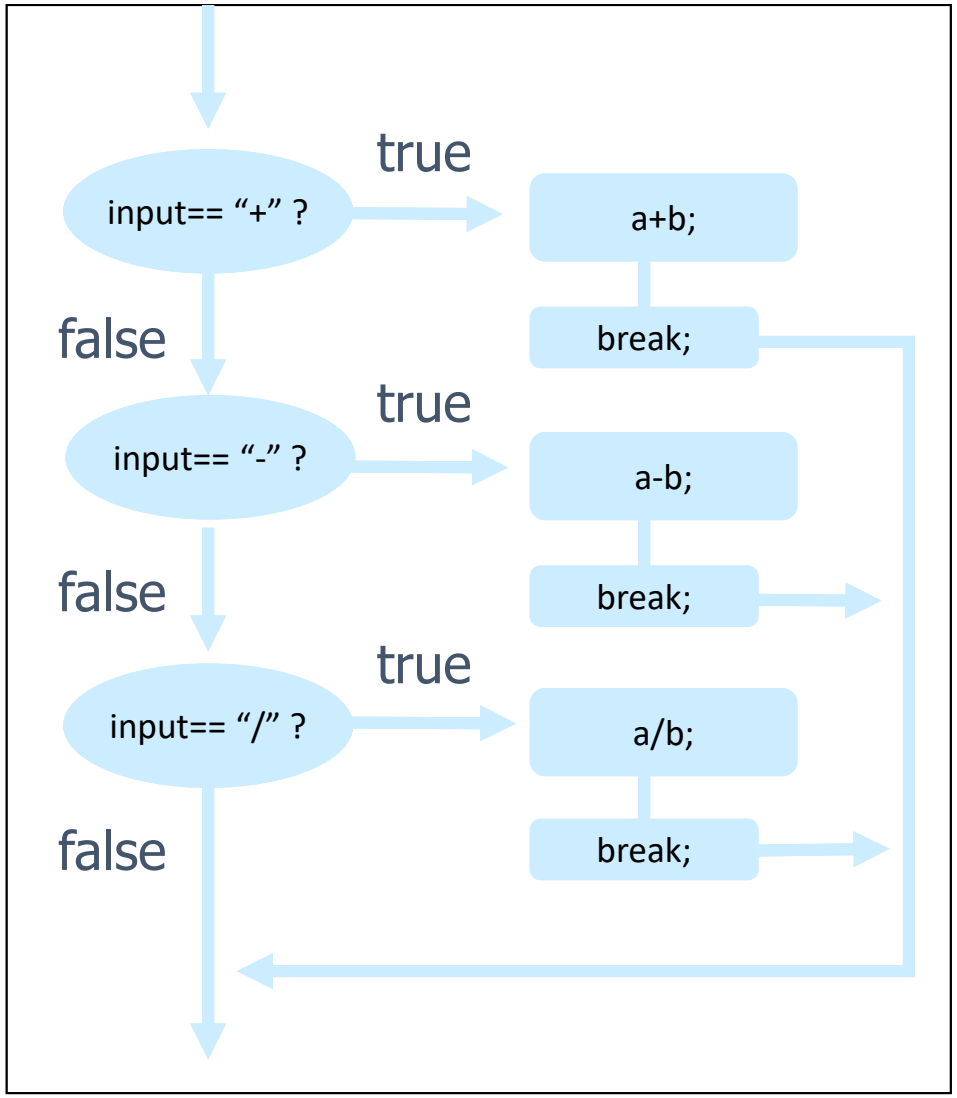
```
switch ( input variable) { // go to the same value
        case value1:        statement(s)1;
                    break; //optional
        case value2:        statement(s)2;
                    break; //optional
            ......
        case valueN:        statement(s)N;
                    break; //optional
        default:    statement(s) ; //optional
 }
```

- switch(variable){}
  - Inside of block{}
  - case comparison target :
- **break;**
  - After processing inside of case
  - Get out of switch block

SWITCH Case Statement with break

```
switch (input) {
    case "+":
    System.out.print(a+b);
            break;
    case "-":
    System.out.print(a-b);
            break;
    case "/":
    System.out.print(a/b);
            break;
}
```

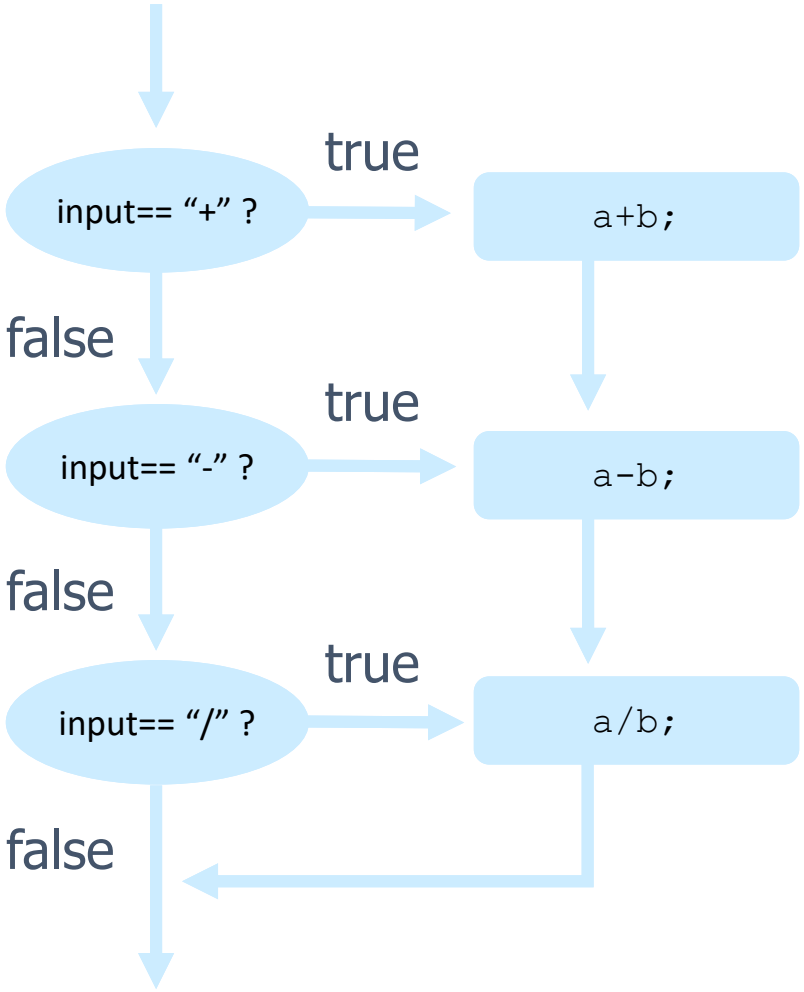
break means that the execution of the whole switch block ends



SWITCH Case Statement Without break

```
switch (input) {
    case "+":
    System.out.print(a+b);
    case "-":
    System.out.print(a-b);
    case "/":
    System.out.print(a/b);
            break;
}
```

If there is no break, the program will continue



Break Statement

- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
- When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.

Practice

1. Make a new project
   - ❑Project name: **Switch_Case**
2. Create a new Class File
   - ❑Class name: **Switch_Case**
3. Coding:

```
public class Switch_Case {
public static void main(String[] args) {
int month = 8;
   String monthString = "";
   switch (month) {
      case 1:  monthString = "January";
               break;
      case 2:  monthString = "February";
               break;
      case 3:  monthString = "March";
               break;
      case 4:  monthString = "April";
               break;
      case 5:  monthString = "May";
               break;
      case 6:  monthString = "June";
               break;
      case 7:  monthString = "July";
               break;
      case 8:  monthString = "August";
               break;
      case 9:  monthString = "September";
               break;
      case 10: monthString = "October";
               break;
      case 11: monthString = "November";
               break;
      case 12: monthString = "December";
               break;
      default: monthString = "Invalid month";
               break;
   }
   System.out.println(monthString);
} }
```

Practice
– Code and Result

Package Explorer
- IF_Statement
- Switch_Case
  - src
    - (default package)
      - Switch_Case.java
  - JRE System Library [jre1.8.0_211]

IF_Statement.java | Calculator.java | Switch_Case.java

```java
public class Switch_Case {
    public static void main(String[] args) {
        int month = 8;
        String monthString = "";
        switch (month) {
            case 1:  monthString = "January";
                     break;
            case 2:  monthString = "February";
                     break;
            case 3:  monthString = "March";
                     break;
            case 4:  monthString = "April";
                     break;
            case 5:  monthString = "May";
                     break;
            case 6:  monthString = "June";
                     break;
            case 7:  monthString = "July";
                     break;
            case 8:  monthString = "August";
                     break;
            case 9:  monthString = "September";
                     break;
            case 10: monthString = "October";
                     break;
            case 11: monthString = "November";
                     break;
            case 12: monthString = "December";
                     break;
            default: monthString = "Invalid month";
                     break;
        }
        System.out.println(monthString);
    }
}
```

Result

Problems | Javadoc | Declaration | Console

```
<terminated> Switch_Case [Java Application] C:\Program Files\Java\jre1.8.0_211\bir
August
```

Summary

- SWITCH Case Statement
  - Multiple Conditions
    - Search equal “case”