TRIMEX COLLEGES Formerly Trimex Institute of Science and Technology	Course Code	**CC3**
	Description	**Intermediate Programming**
	Pre-Requisites	
Department: College of Computer Studies	Week	**5**
Switch Statement in Java		

LEARNING OUTCOMES

At the end of this module, the student will be able to:

- Explain the Switch Statement in Java.
- Demonstrate the example program of Switch Statement.

INTRODUCTION

In the world of programming and coding and generally in information technology, the switch statement is a key feature that you will hear of extensively. The switch statement refers to the selection control mechanism that allows the variable value to change the individual statement's order of the software execution through search.

The switch statement is essential and common in programming languages such as c++, Visual Basic, Java, Ada and many more. This is a feature you need to be conversant with as a programmer more so if you are dealing with program paradigms using statements that change the software's state.

If you are a programmer, the switch statement is a good companion for your coding stint more so when looking at improved performance the same to its readability.

COURSE CONTENT

History Of The Switch Statement

The switch statement concept can be traced back to the early 1950s when programming languages were still in their infancy steps. With developments in the languages, the switch statements too evolved to befit the present state of the languages. At the moment, this feature is extensively used in programming and is a constant in many programming languages to provide a suitable path for coding purposes with its efficiency being a huge determining factor in the outcome.

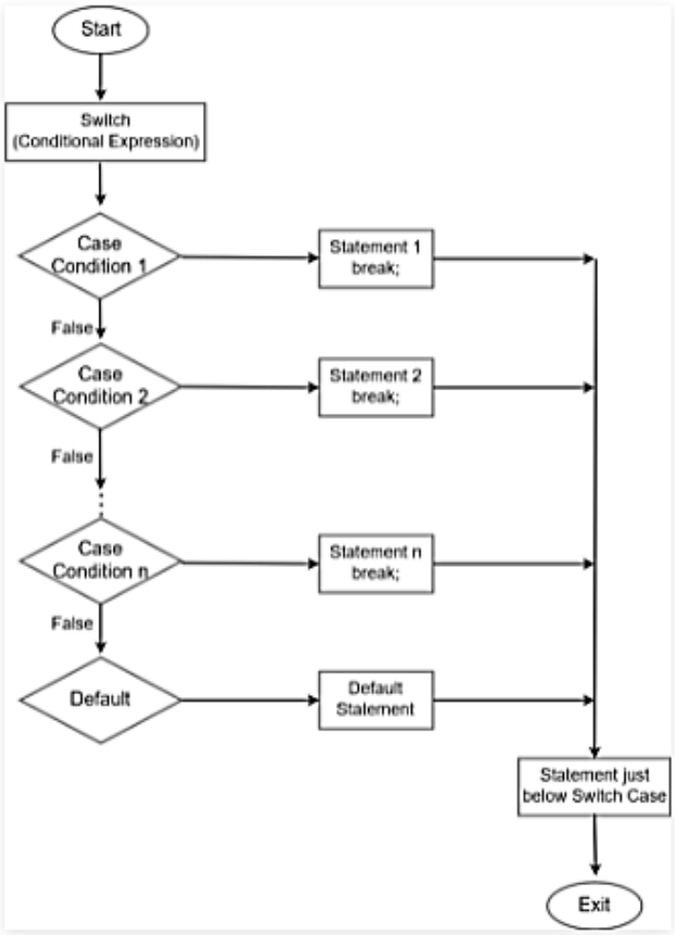
What is switch statement?

The switch statement is a multi-way branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression. Basically, the expression can be byte, short, char, and int primitive data types.

Syntax of Switch-case :

```
// switch statement
switch(expression)
{
   // case statements
   // values must be of same type of expression
   case value1 :
      // Statements
      break; // break is optional

   case value2 :
      // Statements
      break; // break is optional

   // We can have any number of case statements
   // below is default statement, used when none of the
cases is true.
   // No break is needed in the default case.
   default :
      // Statements
}
```

Flow Diagram of Switch-case :



Some Important rules for switch statements :

1. Duplicate case values are not allowed.
2. The value for a case must be of the same data type as the variable in the switch.
3. The value for a case must be a constant or a literal. Variables are not allowed.
4. The break statement is used inside the switch to terminate a statement sequence.
5. The break statement is optional. If omitted, execution will continue on into the next case.
6. The default statement is optional and can appear anywhere inside the switch block. In case, if it is not at the end, then a break statement must be kept after the default statement to omit the execution of the next case statement.

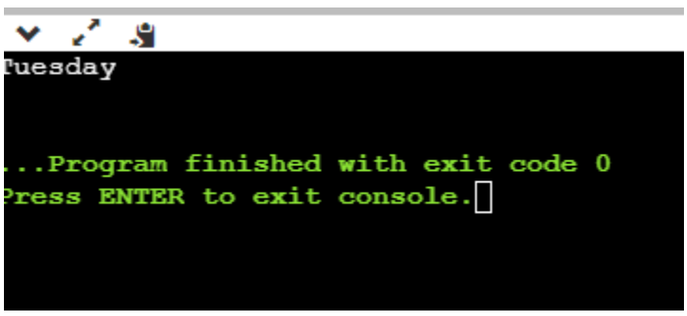
Example 1

Consider the following java program, it declares an int named day whose value represents a day (1-7). The code displays the name of the day, based on the value of the day, using the switch statement.

```
import java.util.Scanner;
 // Java program to demonstrate switch case
// with primitive(int) data type
public class Main
 {
    public static void main(String[] args)
    {
        int day = 2;
        String dayString;
        // switch statement with int data type
        switch (day) {
        case 1:
            dayString = "Monday";
            break;
        case 2:
            dayString = "Tuesday";
            break;
        case 3:
            dayString = "Wednesday";
            break;
        case 4:
            dayString = "Thursday";
            break;
```

```
        case 5:
            dayString = "Friday";
            break;
        case 6:
            dayString = "Saturday";
            break;
        case 7:
            dayString = "Sunday";
            break;
        default:
            dayString = "Invalid day";
        }
        System.out.println(dayString);
    }
}
```

Output:



Omitting the break statement

As break statement is optional. If we omit the break, execution will continue on into the next case. It is sometimes desirable to have multiple cases without break statements between them. For example, consider the updated version of the above program, it also displays whether a day is a weekday or a weekend day.

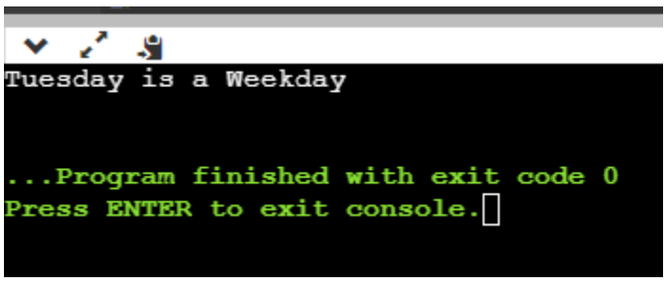
Example

```
import java.util.Scanner;
// Java program to demonstrate switch case
// with multiple cases without break statements
public class Main
 {
    public static void main(String[] args)
    {
        int day = 2;
        String dayType;
        String dayString;

        switch (day) {
        case 1:
            dayString = "Monday";
            break;
        case 2:
            dayString = "Tuesday";
            break;
        case 3:
            dayString = "Wednesday";
            break;
        case 4:
            dayString = "Thursday";
            break;
        case 5:
            dayString = "Friday";
            break;
```

```
case 6:
    dayString = "Saturday";
    break;
case 7:
    dayString = "Sunday";
    break;
default:
    dayString = "Invalid day";
}

switch (day) {
// multiple cases without break statements

case 1:
case 2:
case 3:
case 4:
case 5:
    dayType = "Weekday";
    break;
case 6:
case 7:
    dayType = "Weekend";
    break;

default:
    dayType = "Invalid daytype";
}
```

```
        System.out.println(dayString + " is a " + dayType);
    }
}
```

Output:



Nested Switch Case statements

We can use a switch as part of the statement sequence of an outer switch. This is called a nested switch. Since a switch statement defines its own block, no conflicts arise between the case constants in the inner switch and those in the outer switch.

For example:

```
import java.util.Scanner;
 // Java program to demonstrate
// nested switch case statement
public class Main
 {
   public static void main(String[] args)
   {
      String Program = "BSIT";
      int year = 1;
      switch (year) {
      case 1:
       System.out.println("Courses : Programming  , info Man");
          break;
```

```
        case 2:
          switch (Program) // nested switch
          {
          case "BSIT":
          case "BSCpE":
             System.out.println("Courses : Robotics, AI");
             break;
          case "BSCS":
             System.out.println("Courses : Data Structue");
             break;
          default:
             System.out.println("Courses : Optimization");
          }
       }
    }
}
```

Ouput:

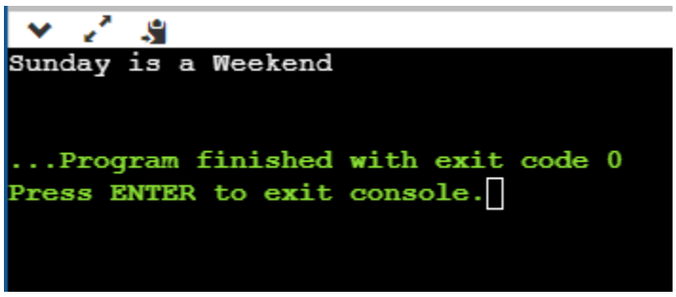
```
Courses : Programming, Info Man


...Program finished with exit code 0
Press ENTER to exit console.
```

FOCUS QUESTIONS

1. How do you write a switch statement in Java?
2. What is switch in Java with example?
3. Is switch case faster than if?

Practice Quiz/Laboratory Activity

Create a program that will display an output:



Answer Program Code:

```
import java.util.Scanner;
public class Main
 {
    public static void main(String[] args)
    {
        int day = 7;
        String dayType;
        String dayString;

        switch (day) {
        case 1:
            dayString = "Monday";
            break;
        case 2:
            dayString = "Tuesday";
            break;
        case 3:
            dayString = "Wednesday";
            break;
        case 4:
            dayString = "Thursday";
            break;
        case 5:
            dayString = "Friday";
            break;
        case 6:
            dayString = "Saturday";
            break;
```

```
            case 7:
               dayString = "Sunday";
               break;
            default:
               dayString = "Invalid day";
            }
            switch (day) {
            case 1:
            case 2:
            case 3:
            case 4:
            case 5:
               dayType = "Weekday";
               break;
            case 6:
            case 7:
               dayType = "Weekend";
               break;

            default:
               dayType = "Invalid daytype";
            }

            System.out.println(dayString + " is a " + dayType);
         }
      }
```

LEARNING ACTIVITIES

A. Discussion Direction:
The instructor will discuss Switch Statement.

B. Demonstration Direction: The instructor will demonstrate on how to create a program using Switch Statement..

C. Individual Activity. Explore and execute what you have learned from the Instructors demonstration.

ASSESSMENT

Recitation

Direction: Question and Answer from instructor to student to evaluate the learning's.

ASSIGNMENT

Read Module 6 found the Learning Paths

RELATED READINGS / REFERENCES
READINGS: https://www.javatpoint.com/java-data-types https://www.javatpoint.com/java-variable https://www.geeksforgeeks.org/

Prepared by:	Approved by:	Noted by:
Mendoza, Analyn R. Faculty	**Carlo Batitis** Program Chair	**Rito A. Camigla Jr., EdD** VP for Academics & Student Services