

# Learning Module

COURSE NUMBER: ITE 103

COURSE TITLE: COMPUTER PROGRAMMING 2

## WEEK 1

### Course Learning Outcomes

Improve previous skills in basic programming in terms of variables, identifiers, and control structures such as branching statements and looping statements.

### Student Learning Outcome

Solve programming problems using branching statements

### Learning Content: Review of Branching Statements in Java

#### ***A. Introduction***

Welcome my dear students to the first week in our course in computer programming 2. In this week will review the previous lessons that we learned from computer programming 1, particularly in the effective use of branching statements in a computer program as part of a solution to a computing problem.

#### ***B. Lesson Content***

As what we have learned in our previous course in Computer Programming 1, there are two types of branching statements namely: if-else statement, and the switch statement. The 'if statement' is a very heavily-used programming statements and it exists in virtually every programming language. At its very basic, it is used to make two different outcomes depending on a certain condition. On the other hand, the switch statement is used to make programs perform several possible outcomes based on a certain value.

**For a more detailed explanation on if-else statement and the switch statement, please watch the two videos included in this module. Furthermore, I also included some sample programs that demonstrates the use of these statements. Please check them out.**

### ***C. Teaching-Learning Activity***

For your activity for this week. Please perform the following task:

#### ***Branching Programming Problem***

It takes approximately 365.25 days for Earth to orbit the Sun — a solar year. We usually round the days in a calendar year to 365. To make up for the missing partial day, we add one day to our calendar approximately every four years. That is a leap year. A leap year is evenly divisible by 4. However, if a year is also evenly divisible by 100, it must also be evenly divisible by 400 in order to be a leap year.

Write a Java program that will accept a integer as a year and will output whether it is a leap year or not.

Sample Output:

Enter a year: 2005

2005 is not a leap year.

Enter a year: 2004

2004 is a leap year.