Coding Exercise: 2.6- Control Flow: The Road Less Traveled [1]

Chapter 2: Control Statements[2]

Jarrian Vince G. Gojar[3]

[1]A coding exercise for Chapter 2of the Study Guide on the course Object-Oriented Programming.
[2]This chapter introduces the basic concepts of control statements in Java programming language.
[3]https://github.com/godkingjay

Sorsogon State University - Bulan Campus

The Scanner Class

The "Scanner" class can be used to read input from the user. The following exercises are designed to help you practice using control statements in Java. You can use the "Scanner" class to read input from the user and test your solutions.

```java
// ScannerExample.java
import java.util.Scanner;

public class ScannerExample {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter a number: ");
    int number = scanner.nextInt();
    System.out.println("You entered: " + number);

    scanner.close();
  }
}
```

Code 1: Simple Implementation of the Scanner Class

In the above code, we import the "Scanner" class by using the statement "import java.util.Scanner;". We create an instance of the "Scanner" class by using the statement "Scanner scanner = new Scanner(System.in);". We read an integer input from the user by using the statement "int number = scanner.nextInt();". We print the input number by using the statement "System.out.println("You entered: " + number);". Finally, we close the scanner by using the statement "scanner.close();".

The following are used to read different types of input from the user:

- "scanner.nextBoolean()" to read a boolean value.
- "scanner.nextByte()" to read a byte value.
- "scanner.nextShort()" to read a short value.
- "scanner.nextInt()" to read an integer value.
- "scanner.nextLong()" to read a long value.
- "scanner.nextFloat()" to read a float value.
- "scanner.nextDouble()" to read a double value.
- "scanner.nextLine()" to read a string value.
- "scanner.next()" to read a single word.

Coding Exercises

Instructions: Write a program that solves the following problems. Submit your code to the Google Drive folder provided by the instructor.

Note 1: You have to answer 3 out of the 5 exercises.

Note 2: You have to answer Exercise 1, Exercise 5, and one of the remaining exercises. (e.g., Exercise 2, Exercise 3, Exercise 4) for a total of 3 exercises.

Note 3: In the following exercises, you will use the “Scanner” class to read input from the user.

Exercise 1: Find the Largest Number (Conditional Statements)

In this exercise, you will write a Java program to find the largest of three numbers entered by the user. You should use the “Scanner” class to read input from the user and conditional statements to compare the three numbers and determine the largest number. Finally, you should print the largest number to the console. If the numbers are equal, you should print a message saying "There is no largest number".

1. With the use of the “Scanner” class, read three numbers from the user. and store them in variables “num1”, “num2”, and “num3”.

2. Use conditional statements to compare the three numbers and determine the largest number.

   (a) If “num1” is greater than “num2” and “num1” is greater than “num3”, then “num1” is the largest number and print "The largest number is: num1".

   Example: If “num1” is 5, “num2” is 3, and “num3” is 4, then the output should be "The largest number is: 5".

   (b) If none of the above conditions are true, then the numbers are equal. Thus, print a message saying "There is no largest number".

   Example: If “num1” is 5, “num2” is 5, and “num3” is 5, then the output should be "There is no largest number".

Exercise 2: Print Multiplication Table (Iteration Statements)

In this exercise, you will write a Java program to print the multiplication table of a number entered by the user. You should use the “Scanner” class to read input from the user and iteration statements to print the multiplication table of the entered number. Finally, you should print the multiplication table to the console.

1. With the use of the “Scanner” class, read a number from the user and store it in a variable “number”.

2. Use a “for” loop to iterate from 1 to 10.

   (a) Inside the loop, calculate the product of “number” and the loop variable and store it in a variable “result”.

   (b) Print the multiplication table in the format "number * loop variable = result".

   Example:
   If the user enters 5, then the output should be:
   5 * 1 = 5
   5 * 2 = 10
   5 * 3 = 15
   5 * 4 = 20
   5 * 5 = 25
   5 * 6 = 30
   5 * 7 = 35

5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

Exercise 3: Factorial of a Number (Iteration Statements)

In this exercise, you will write a Java program to calculate the factorial of a number entered by the user. You should use the "Scanner" class to read input from the user and iteration statements to calculate the factorial of the entered number. Finally, you should print the factorial to the console.

A factorial of a non-negative integer "n" is the product of all positive integers less than or equal to "n". It is denoted by "n!".

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1 \tag{1}$$

The formula above shows how to calculate the factorial of a number.

Example: The factorial of 5 is calculated as follows:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120 \tag{2}$$

In this exercise, you will calculate the factorial of a number entered by the user.

1. With the use of the "Scanner" class, read a number from the user and store it in a variable "number".
2. Initialize a variable "factorial" to 1.
3. Use a "for" loop to iterate from 1 to "number".
   (a) Inside the loop, multiply the loop variable by "factorial" and store the result in "factorial".
   (b) Print the factorial of the entered number.

   Example:
   If the user enters 5, then the output should be:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120 \tag{3}$$

   Thus, the factorial of 5 is 120.

Exercise 4: Fibonacci Series (Iteration Statements)

In this exercise, you will write a Java program to print the Fibonacci series up to a specified number of terms entered by the user. You should use the "Scanner" class to read input from the

user and iteration statements to calculate and print the Fibonacci series. Finally, you should print the Fibonacci series to the console.

The Fibonacci series is a series of numbers in which each number is the sum of the two preceding ones, usually starting with 0 and 1. The sequence goes 0, 1, 1, 2, 3, 5, 8, 13, 21, and so on.

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2} \text{ for } n > 1 \tag{4}$$

The formula above shows how to calculate the Fibonacci series.

Example: The Fibonacci series up to 10 terms is calculated as follows:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34 \tag{5}$$

In this exercise, you will calculate and print the Fibonacci series up to a specified number of terms entered by the user.

1. With the use of the "Scanner" class, read the number of terms from the user and store it in a variable "terms".
2. Initialize two variables "firstTerm" and "secondTerm" to 0 and 1 respectively.
3. Print the first two terms of the Fibonacci series.
4. Use a "for" loop to iterate from 1 to "terms".
   (a) Inside the loop, calculate the next term of the Fibonacci series by adding "firstTerm" and "secondTerm" and store it in a variable "nextTerm".
   (b) Print the next term of the Fibonacci series.
   (c) Update the values of "firstTerm" and "secondTerm" to the previous two terms of the Fibonacci series.

   Example:
   If the user enters 10, then the output should be:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34 \tag{6}$$

   Thus, the Fibonacci series up to 10 terms is 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

Exercise 5: Sum of Numbers (Jump Statements)

In this exercise, you will write a Java program to calculate the sum of numbers entered by the user. You should use the "Scanner" class to read input from the user and jump statements to control the flow of the program. Finally, you should print the sum of the numbers to the console.

1. With the use of the "Scanner" class, read numbers from the user until the user enters a negative number. Store the numbers in a variable "number" and the sum of the numbers in a variable "sum".

2. Use a "while" loop to read numbers from the user until the user enters a negative number.

   (a) Inside the loop, check if the value of "number" is negative. If it is, break out of the loop.

   (b) Add the value of "number" to the sum of the numbers.

   (c) Continue reading numbers from the user.

   Example:
   If the user enters 5, 10, 15, -1, then the output should be:
   The sum of the numbers is: 30

   If the user enters 10, 20, 30, 40, 50, -1, then the output should be:
   The sum of the numbers is: 150

Submission of Coding Exercises

Instructions:

1. Go to the Google Drive folder provided by the instructor:

   For BSCS 2-1:
   https://drive.google.com/drive/folders/1c56xFCJgFh6FWQQ4iZ-UuKKcWioF8pgs?usp=sharing

   For BSCS 2-2:
   https://drive.google.com/drive/folders/1jANc3o6atOYbHyoJZ6b-j-nDlTknEiu-?usp=sharing

2. Inside the folder, create another folder for your group with the following format:

   Group Number - LastName1_FirstName1, LastName2_FirstName2
   Example: **Group 1 - Doe_John, Smith_Jane**

3. Inside the sub-folder, create another folder with the name:

   Chapter 2- Coding Exercise 2.6- Control Flow: The Road Less Traveled

4. Inside the folder, upload the file of your submission.

   Fill in the template provided in the following link and upload it inside the folder:
   https://docs.google.com/document/d/1sctvVLgpPSVnXN82k6LsOPSPApKp2rV0/edit?usp=drive_link&ouid=112709378145681657270&rtpof=true&sd=true

5. The activity must be submitted **on or before October 11, 2024**.

6. Late submissions will not be accepted.