CSE 215L: Programming Language II Lab
Faculty: Silvia Ahmed, Sec – 12, 13

Lab 04 – Summer 2020

Objective:
After today's lab, the students should be able:
- To define methods with formal parameters
- To invoke methods with actual parameters (i.e., arguments)
- To define methods with or without a return value
- To use method overloading and understand ambiguous overloading

Defining a Method

A method definition consists of its method name, parameters, return value type, and body.

The syntax for defining a method is as follows:

```
modifier returnValueType methodName(list of parameters) {
// Method body;
}
```

Task – 1

(*Math: pentagonal numbers*) A pentagonal number is defined as $n(3n–1)/2$ for $n = 1, 2, . . .,$ and so on. Therefore, the first few numbers are 1, 5, 12, 22, ... Write a method with the following header that returns a pentagonal number:

public static int getPentagonalNumber(**int** n)

Write a test program that uses this method to display the first 100 pentagonal numbers with 10 numbers on each line.

Task – 2

(*Sum the digits in an integer*) Write a method that computes the sum of the digits in an integer. Use the following method header:

public static int sumDigits(**long** n)

For example, **sumDigits(234)** returns **9** (2 + 3 + 4). (*Hint*: Use the **%** operator to extract digits, and the **/** operator to remove the extracted digit. For instance, to extract 4 from 234, use **234 % 10** (= 4). To remove 4 from 234, use **234 / 10** (= 23). Use a loop to repeatedly extract and remove the digit until all the digits are extracted. Write a test program that prompts the user to enter an integer and displays the sum of all its digits.

Task – 3

(*Palindrome integer*) Write the methods with the following headers

// Return the reversal of an integer, i.e., reverse (456) returns 654
public static int reverse(**int** number)
// Return true if number is a palindrome
public static boolean isPalindrome(**int** number)

Use the **reverse** method to implement **isPalindrome**. A number is a palindrome if its reversal is the same as itself. Write a test program that prompts the user to enter an integer and reports whether the integer is a palindrome.

Task – 4
(*Sort three numbers*) Write a method with the following header to display three numbers in increasing order:

public static void displaySortedNumbers(**double** num1, **double** num2, **double** num3)

Write a test program that prompts the user to enter three numbers and invokes the method to display them in increasing order.

Task – 5
(*Financial application: compute the future investment value*) Write a method that computes future investment value at a given interest rate for a specified number of years. The future investment is determined using the following formula.

$$\text{futureInvestmentValue} = \text{investmentAmount} * (1 + \text{monthlyInterestRate})^{\text{numberOfYears} * 12}$$

Use the following method header:
public static double futureInvestmentValue(**double** investmentAmount, **double** monthlyInterestRate, **int** years)
For example, **futureInvestmentValue(10000, 0.05/12, 5)** returns **12833.59**.
Write a test program that prompts the user to enter the investment amount (e.g.,1000) and the interest rate (e.g., 9%) and prints a table that displays future value for the years from 1 to 30, as shown below:

```
The amount invested: 1000
Annual interest rate: 9
Years Future Value
1 1093.80
2 1196.41
...
29 13467.25
30 14730.57
```

Task – 6
(*Display matrix of 0s and 1s*) Write a method that displays an *n*-by-*n* matrix using the following header:

public static void printMatrix(**int** n)

Each element is 0 or 1, which is generated randomly. Write a test program that prompts the user to enter **n** and displays an *n*-by-*n* matrix. Here is a sample run:

```
Enter n: 3
0 1 0
0 0 0
1 1 1
```

Task – 7

(*The* **MyTriangle** *class*) Create a class named **MyTriangle** that contains the following two methods:

/** Return true if the sum of any two sides is
* greater than the third side. */
public static boolean isValid(**double** side1, **double** side2, **double** side3)
/** Return the area of the triangle. */
public static double area(**double** side1, **double** side2, **double** side3)

Write a test program that reads three sides for a triangle and computes the area if the input is valid. Otherwise, it displays that the input is invalid. The formula for computing the area of a triangle is

$$s = (\text{side1} + \text{side2} + \text{side3})/2;$$

$$\text{area} = \sqrt{s(s - \text{side1})(s - \text{side2})(s - \text{side3})}$$

Task – 8

(*Convert milliseconds to hours, minutes, and seconds*) Write a method that converts milliseconds to hours, minutes, and seconds using the following header:

public static String convertMillis(**long** millis)

The method returns a string as ***hours*:*minutes*:*seconds***. For example, **convertMillis(5500)** returns a string 0:0:5, **convertMillis(100000)** returns a string **0:1:40**, and **convertMillis(555550000)** returns a string **154:19:10**.

Task – 9

(*Geometry: area of a regular polygon*) A regular polygon is an *n*-sided polygon in which all sides are of the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). The formula for computing the area of a regular polygon is

$$Area = \frac{5 * s2}{4 * \tan\left(\frac{\pi}{5}\right)}$$

Write a method that returns the area of a regular polygon using the following header:

public static double area(**int** n, **double** side)

Write a main method that prompts the user to enter the number of sides and the side of a regular polygon and displays its area. Here is a sample run:

```
Enter the side: 5.5
The area of the pentagon is 52.04444136781625
```