

King Saud University
College of Computer & Information Science
CSC111 – Tutorial08
Methods
All Sections

Objectives:

Student should learn how to:

- 1- To define methods with formal parameters.
- 2- To invoke methods with actual parameters (i.e., arguments).
- 3- To define methods with a return value.
- 4- To define methods without a return value.
- 5- To pass arguments by value.
- 6- To use method overloading and understand ambiguous overloading
- 7- To determine the scope of variables.

Exercise 1

- 1) Given the following method, point out the following:
 - a) Modifier
 - b) Return type
 - c) Method header
 - d) Method signature
 - e) Method name
 - f) Parameters
 - g) Method body

```

public static int max(int num1, int num2) {
    int result;

    if (num1 > num2)
        result = num1;
    else
        result = num2;

    return result;
}

```

2) What is the output of the following code:

```

public class AdditionTest {
    public static int add(int x, int y){
        x++;
        return x + y;
    }
    public static void main(String[] args) {
        int a = 5, x = 1;
        int y = add(a, add(x, 3 * a));
        System.out.println("a = " + a + ", x = " + x + ", y = " + y);
    }
}

```

3) Write only method headers (not the bodies) for the following methods:

- a) Return a sales commission, given the sales amount (int) and the commission rate (double).
- b) Display the calendar for a month, given the month (String) and year (int).
- c) Return a square root of a number.
- d) Test whether a number is even, and returning **true** if it is.
- e) Display a message a specified number of times.

4) Identify and correct the errors in the following program:

```
1  public class Test {
2      public static method1(int n, m) {
3          n += m;
4          method2(3.4);
5      }
6
7      public static int method2(int n) {
8          if (n > 0) return 1;
9          else if (n == 0) return 0;
10         else if (n < 0) return -1;
11     }
12 }
```

5) Identify and correct the errors in the following program:

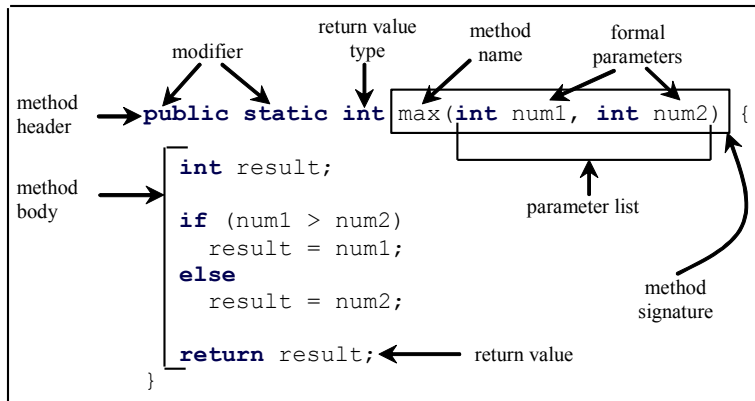
```
1  public class Test {
2      public static void main(String[] args) {
3          nPrintln(5, "Welcome to Java!");
4      }
5
6      public static void nPrintln(String message, int n) {
7          int n = 1;
8          for (int i = 0; i < n; i++)
9              System.out.println(message);
10     }
11 }
```

6) Show the result of the following program:

```
public class Test {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 6) {  
            method1(i, 2);  
            i++;  
        }  
    }  
  
    public static void method1(  
        int i, int num) {  
        for (int j = 1; j <= i; j++) {  
            System.out.print(num + " ");  
            num *= 2;  
        }  
  
        System.out.println();  
    }  
}
```

Solution

1)



2)

```
a = 5, x = 1, y = 23
```

3)

- a) `public static double comm(int sales, double rate)`
- b) `public static void displayCal(String month, int year)`
- c) `public static double sqrt(double num)`
- d) `public static boolean isEven(int num)`
- e) `public static void displayMsg(int numOfTimes)`

4)

```
1 public class Test { void
2     public static method1(int n, m) { int
3         n += m;
4         method2(3.4);
5     }
6
7     public static int method2(int n) {
8         if (n > 0) return 1;
9         else if (n == 0) return 0;
10        else if (n < 0) return -1;
11    }
12 }
```

Annotations in the code:

- A red circle around `void` with an arrow pointing to it from the label `void`.
- A red circle around `int` with an arrow pointing to it from the label `int`.
- A red circle around `3.4` with an arrow pointing to it from the label `Has to be an integer`.
- A red circle around the closing brace of the class `}` with an arrow pointing to it from the label `No main method`.

5)

```
1  public class Test {
2      public static void main(String[] args) {
3          nPrintln(5, "Welcome to Java!");
4      }
5
6      public static void nPrintln(String message, int n) {
7          int n = 1;
8          for (int i = 0; i < n; i++)
9              System.out.println(message);
10     }
11 }
```

order of arguments is wrong

n defined before

6)

```
2
2 4
2 4 8
2 4 8 16
2 4 8 16 32
2 4 8 16 32 64
```

Exercise 2

Write a program to display three numbers in increasing order. Use a method with the following header:

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

Solution

```
import java.util.Scanner;
public class SortNums {
    public static void main(String[] args) {
        // Prompt the user to enter three double values
        Scanner input = new Scanner(System.in);

        // Enter three numbers
        System.out.print("Enter three values: ");
        double num1 = input.nextDouble();
        double num2 = input.nextDouble();
        double num3 = input.nextDouble();

        // Invoke the displaySortedNumbers method to display the
        // numbers in increasing order
        displaySortedNumbers(num1, num2, num3);
    }

    public static void displaySortedNumbers(
        double num1, double num2, double num3) {
        // Write the code to implement this method
        if (num1 > num2) {
            double temp = num1;
            num1 = num2;
            num2 = temp;
        }

        if (num2 > num3) {
            double temp = num2;
            num2 = num3;
            num3 = temp;
        }

        if (num1 > num2) {
            double temp = num1;
            num1 = num2;
            num2 = temp;
        }

        System.out.println("The sorted numbers are "
            + num1 + " " + num2 + " " + num3);
    }
}
```

Done...