CHOO TIAN LONG 24068668

WIX1002 Fundamentals of Programming
Tutorial 6 Java Methods

1. Write statements for each of the following
    a. Define a static method that returns the maximum number from 3 integer parameters.

```
public static int MaxNumber(int num1, int num2, int num3){
    return Math.max(num1, Math.max(num2, num3));
}
```

    b. Define a static method that that determine whether the given integer is a square number.

```
public static boolean isSquare(int number){
    return Math.pow((int)Math.sqrt(number), 2) == number;
  }
```

    c. Define a static method that use to compute combination function C(n,k). C(n,k) gives the number of different k-element subsets that can be found in a given set of n elements. C(n,k) = n! / (k! (n-k)!

```
public static int binomialCoefficient(int n, int k){
    // k cant be greater than n
    if (k > n) {
       return 0;
    }
    // nC0 = 1 and if n = 10, r = 10 10C10= 1
    if (k == 0 || k == n){
       return 1;
    }
    // Recursive add value
    return binomialCoefficient(n-1, k-1) + binomialCoefficient(n-1, k);
  }
```

d. Define a static method that used to determine whether the parameter is a pentagonal number. A pentagonal number is a figurate number that extends the concept of triangular and square numbers to the pentagon. Pn = ½ n(3n -1)

```
public static boolean isPentagonal(int num){
    int Pn = 1;
    // Checking through every pentagonal number
    for (int n = 1; Pn <= num; n++) {
       Pn =  (n * ((3*n) - 1)) / 2;
       if(Pn == num){
          return true;
       }
    }
    return false;
  }
```

e. Define a static method that displays the number of letters and the number of digits of a String parameter.

```
public static void string_length(String sentence){
    int index = 0, letter = 0, digit = 0;
    while (index < sentence.length()){
       char currentChar = sentence.charAt(index);
       if (Character.isLetter(currentChar)) {
          letter++;
       } else if (Character.isDigit(currentChar)){
          digit++;
       }

       index++;
    }
    System.out.println("The number of letters of the string: "+ letter);
    System.out.println("The number of digits of the string: "+ digit);
  }
```

f. Define a static void method that generates 10 random numbers within 0 to 100 to the method's parameter. The random numbers can be accessed by the main method.

```
public static void RandomNumbers(int[] numbers){
    Random rd = new Random();

    for (int i = 0; i < numbers.length; i++) {
       numbers[i] = rd.nextInt(101);

    }
  }
```

g. Define a static void method that returns the area and the circumference of a circle given the argument is radius. Area = πr^2 and Circumference = $2 \pi r$.

```
public static void circle(double r, double[] ac){
    ac[0] = Math.PI * (r*r);
    ac[1] = 2 * Math.PI * r;
  }
```

h. Define a static method that generate random number within 0 – 10. The method will return the first random number that generate twice.

```
public static int isDuplicate(boolean[] numbers){
    Random rd = new Random();
    int number;


    while(true){
        number = rd.nextInt(11);
//          System.out.print(number + " "); // data validation


        if(numbers[number]){
            return number; // return once found the first duplicate
        } else{
            numbers[number] = true; // mark the number as seen
        }
    }


}
```

2. Write a program that consists of a method that can display three numbers in decreasing order.

```
import java.util.Scanner;
public class T6Q2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int num;
        int numbers[] = new int[3];

        System.out.print("Enter three numbers: ");

        for (int i = 0; i < 3; i++) {
            num = sc.nextInt();
            numbers[i] = num;
        }

        descend(numbers);
    }
    public static void descend(int[] numbers){
        for (int i = 0; i < numbers.length - 1; i++) {
            for (int j = 0; j < numbers.length - i - 1; j++) {
                if (numbers[j] < numbers[j+1]) {
                    int temp = numbers[j];
                    numbers[j] = numbers[j+1];
                    numbers[j+1] = temp;
                }
            }
        }

        for (int i = 0; i < numbers.length; i++) {
             if (i != numbers.length - 1) {
                 System.out.print(numbers[i] + ", ");
            }else{
```

```
            System.out.println(numbers[i]);
        }


    }
  }
}
```