# COP-2210
# Computer Programming I

Instructor: Dr. Antonio Hernandez

Text: Big Java: Early Objects, Interactive Edition, 6th Edition

# PRACTICE

**Program 26_08:**

Write a printError method that displays an error message. No value to be passed or returned. Test the method by invoking it from the class constructor.

# PRACTICE

**Program 26_09:**

Write a **minMax** method that has two integers passed to it. The method prints the value of the minimum and the value of the maximum. Test your method.

# PRACTICE

**Program 26_10:**

Write a **getPositiveInt** method that asks the user to enter a positive integer number and then returns it. If the number is 0 or negative, the method will return -1. Test your method.

# PRACTICE

**Program 26_11:**

Write a **factorial** method that calculates and returns the factorial of a number passed as a parameter. Test your method by printing the factorials of numbers between 1 and 10.

# PRACTICE

**Program 26_12:**

Write and test an **allTogether** method that

1) Gets two positive integers from the user.
2) Prints an error if any of the numbers is not greater than 0 and halts the execution of the method.
3) Calculates the factorial of each of the numbers.
4) Prints in increasing order the two factorials.

# Program26_12: *Solution*

```java
import java.util.*;

public class Prog26_12
{
    public static void main(String[] args)
    {
        new Prog26_12();
    }

    public Prog26_12()
    {
        allTogether();
    }

    public void printError()
    {
        System.out.println("Error!");
    }
```

```java
    public int getPositiveInt()
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a positive integer: ");

        int n = in.nextInt();

        if (n<=0) return -1;
        else return n;
    }

    public void minMax(int a, int b)
    {
        if (a<b) System.out.println(a + ", " + b);
        else System.out.println(b + ", " + a);
    }
```

# Program26_12: *Solution*

```
    public int factorial(int n) {
        int fact = 1;

        for (int i=1; i<=n; i++)
        {
            fact *= i;
        }

        return fact;
    }

    public void allTogether()
    {
        int n1 = getPositiveInt();
        int n2 = getPositiveInt();

        if (n1 == -1 || n2 == -1) printError();
        else minMax(factorial(n1), factorial(n2));
    }
}
```

# Defining Your Own Classes

## 27. Methods and Arrays

# Passing arrays to functions

*Called function:*

<return type> <f. name> ( … <array type> [ ] <array name> …);

*Calling function:*

… <f. name> ( … <array name> …) …

# Passing arrays to functions: example

*Called function:*

```
int myFunction ( int[ ] q ) { … }
```

*Calling function:*

```
int[ ] a = new int [5];

int  v =  myFunction ( a ) ;
```

# Passing arrays to functions: *Try it yourself*

```java
import java.util.*;

public class Prog27_01
{
    public static void main(String[] args)
    {
        new Prog27_01();
    }

    public Prog27_01()
    {
        Random rnd = new Random();

        int[] arr = new int[5];
        for (int i = 0; i < arr.length; i++)
        {
            arr[i] = rnd.nextInt(10);
            System.out.print(arr[i] + " ");
        }
        System.out.println("\nSum of array elements = " +
                                    arraySum(arr));
    }
```

```java
    public int arraySum(int[] a)
    {
        int sum = 0;
        for(int n : a)
        {
            sum += n;
        }

        return sum;
    }
}
```

# PRACTICE

**Program 27_02:**

Write a printArray method that displays an array passed as a parameter. Test your method.



# Program27_02: *Solution*

```java
import java.util.*;

public class Prog27_02
{
    public static void main(String[] args)
    {
        new Prog27_02();
    }

    public Prog27_02()
    {
        Random rnd = new Random();

        int[] arr = new int[5];
        int size = 5;
        for (int i = 0; i < size; i++)
        {
            arr[i] = rnd.nextInt(10);
        }

        printArray(arr);
    }
```

```java
    public void printArray(int[] a)
    {
        for(int i = 0; i<a.length; i++)
        {
            System.out.print(a[i] + " ");
        }

        System.out.println();
    }
}
```

# Passing arrays to functions: *Try it yourself*

```
import java.util.*;

public class Prog27_03
{
    public static void main(String[] args)
    {
        new Prog27_03();
    }

    public Prog27_03()
    {
        Random rnd = new Random();

        int[] arr = new int[20];
        int size = 5;
        for (int i = 0; i < size; i++)
        {
            arr[i] = rnd.nextInt(10);
        }

        printArray(arr);
    }
```

```
    public void printArray(int[] a)
    {
        for(int i = 0; i<a.length; i++)
        {
            System.out.print(a[i] + " ");
        }

        System.out.println();
    }
}
```

# Passing arrays to functions: *Try it yourself*

```java
import java.util.*;

public class Prog27_04
{
    public static void main(String[] args)
    {
        new Prog27_04();
    }

    public Prog27_04()
    {
        Random rnd = new Random();

        int[] arr = new int[20];
        int size = 5;
        for (int i = 0; i < size; i++)
        {
            arr[i] = rnd.nextInt(10);
        }

        printArray(arr, size);
    }
```

```java
    public void printArray(int[] a, int total)
    {
        for(int i = 0; i<total; i++)
        {
            System.out.print(a[i] + " ");
        }

        System.out.println();
    }
}
```

# PRACTICE

**Program 27_05:**

Write a **maxValue** method that returns the greatest number stored in an array passed as a parameter. The total number of elements in the array will be passed also. Test your method.

# Returning arrays: *Try it yourself*

```
public class Prog27_06
{

    public static void main(String[] args)
    {
        new Prog27_06();
    }

    public Prog27_06()
    {
        char[] arr;

        arr = fillArray();

        for (char c : arr)
        {
            System.out.print(c + " ");
        }

        System.out.println();
    }
```

```
    public char[] fillArray()
    {
        char[] a = {'a', 'e', 'i', 'o', 'u'};

        return a;
    }
}
```

# PRACTICE

**Program 27_07:**

Write a fillArray method that returns an integer array populated with random numbers. Test your method.

-The entire array will be populated
-Each random number will be in [0, 30]

# Returning arrays: *Try it yourself*

```java
public class Prog27_08
{

    public static void main(String[] args)
    {
        new Prog27_08();
    }

    public Prog27_08()
    {
        char[] arr = new char[5];

        fillArray(arr);

        for (char c : arr)
        {
            System.out.print(c + " ");
        }

        System.out.println();
    }
```

```java
    public void fillArray(char[] a)
    {
        a[0] = 'A';
        a[1] = 'E';
        a[2] = 'I';
        a[3] = 'O';
        a[4] = 'U';
    }
}
```

# PRACTICE

**Program 27_09:**

Write a fillArray method that populates an integer array with random numbers. Pass the array as a parameter. Test your method.

# PRACTICE

**Program 27_10:**

Write a **fillArrays** method that populates two arrays with random numbers. One is an integer array with values in [0, 10] and the other is a double array with values in [0, 1).

```
void fillArrays(int[] a, double[] b)
```

Test your method.