COP-2210
Computer Programming I

Instructor: Dr. Antonio Hernandez

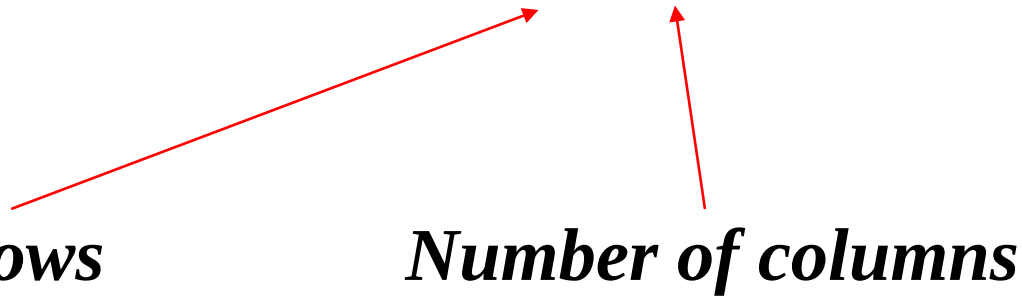
Text: Big Java: Early Objects, Interactive Edition, 6th Edition

Multidimensional Arrays

Declaration of a two dimensional array:

<type>[ ][ ] <name> = new <type>[<number>][<number>];

Example: int [ ][ ] myArray = new int [3][4];



(3-dim. arrays and higher dimensional arrays are declared similarly)

Two-dim. Arrays: *Try it yourself*

```
public class Prog24_11
{
   public static void main(String[] args)
   {
      int [ ][ ] matrix = new int [3][4];

      for (int i=0; i<3; i++)
      {
         for(int j=0; j<4; j++)
         {
            matrix[i][j] = 1;
            System.out.print(matrix[i][j] + " ");
         }

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

Two-dim. Arrays: *Try it yourself*

```
public class Prog24_12
{
   public static void main(String[] args)
   {
      int [ ][ ] matrix = new int [3][4];

      for (int i=0; i<matrix.length; i++)
      {
         for(int j=0; j<matrix[0].length; j++)
         {
            matrix[i][j] = 1;
            System.out.print(matrix[i][j] + " ");
         }

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

matrix.length: no. of rows

matrix[0].length: no. of columns

PRACTICE

Program 24_13:

Write a Java program that declares and populates the identity matrix of size n, I_n, where n will be entered by the user.

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$I_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Program 24_13: *Solution*

```
public class Prog24_13 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Enter matrix size: ");
        int size = in.nextInt();

        int[][] matrix = new int[size][size];

        for (int i=0; i<matrix.length; i++) {
            for (int j=0; j<matrix[0].length; j++) {
                if (i == j) matrix[i][j] = 1;
                else matrix[i][j] = 0;

                System.out.printf("%4d", matrix[i][j]);
            }
            System.out.println();
        }
    }
}
```

Two-dim. Arrays Initialization: *Try it yourself*

```
//Prog24_14: initializing a 2-dim array

public class Prog24_14
{
   public static void main(String[] args)
   {
      int [ ][ ] matrix = {{1, 2, 3 ,4}, {5, 6, 7, 8}, {9, 10, 11, 12}};

      for (int i=0; i<matrix.length; i++)
      {
         for (int j=0; j<matrix[0].length; j++)
         {
            System.out.printf("%4d", matrix[i][j]);
         }

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

PRACTICE

Program 24_15:

Write a Java program that fills a 7x8 matrix with random numbers in [0, 99] and calculates the sum of the numbers.

Program 24_15: *Solution*

```
public class Prog24_15 {
    public static void main(String[] args) {
        int [ ][ ] matrix = new int[7][8];

        Random rnd = new Random();
        int sum = 0;
        for (int i=0; i<matrix.length; i++) {
            for(int j=0; j<matrix[0].length; j++) {
                matrix[i][j] = rnd.nextInt(100);
                System.out.printf("%4d", matrix[i][j]);

                sum += matrix[i][j];
            }

            System.out.println();
        }

        System.out.println("Sum = " + sum);
    }
}
```

PRACTICE

Program 24_16:

Write a Java program that fills a 3x4 matrix with random numbers in [0, 10] and calculates the average of the greatest elements in each row.

$$\begin{bmatrix} 0 & 3 & 2 & 7 \\ 6 & 0 & 3 & 1 \\ 5 & 1 & 4 & 8 \end{bmatrix}$$

→ Max = 7

→ Max = 6

→ Max = 8

Ave = (7+6+8)/3 = 7

Program 24_16: *Solution*

```
public class Prog24_16 {
    public static void main(String[] args) {
        int [ ][ ] matrix = new int[3][4];

        //populate the matrix with random numbers in [0, 10] here

        double ave = 0;
        for (int i=0; i<matrix.length; i++) {
            int max = matrix[i][0];
            for(int j=1; j<matrix[0].length; j++) {
                if (matrix[i][j] > max) max = matrix[i][j];
            }

            System.out.println("Max = " + max);
            ave += max;
        }

        System.out.printf("Average = %5.2f\n", ave/3);
    }
}
```

Array Lists

25. java.util.ArrayList

ArrayList: declaration

java.util.ArrayList:

- format:

ArrayList<*data type*> <*name*>;

<*name*> = *new ArrayList<>();*

or

ArrayList<data type> <name> = new ArrayList<>();

ArrayList: declaration

Example:

```
ArrayList<Double> bills;
bills = ArrayList<>();
```

or

```
ArrayList<Double> bills  = new ArrayList<>();
```

ArrayLists: access via methods

```
//ArrayList: generic class
ArrayList<Integer> numbers = new ArrayList<>();

//list creation
for(int i=0; i < 10; i++)
        numbers.add(rnd.nextInt(100));

System.out.println("List of numbers:");
System.out.println(numbers);

Scanner in = new Scanner(System.in);
System.out.print("Please, enter an int value: ");
int x = in.nextInt();

if (numbers.contains(x)) System.out.println("Found!");
else System.out.println("Not found!");
```

ArrayList: *Try it yourself*

```
import java.util.ArrayList;

public class Prog25_01
{
   public static void main(String[] args)
   {
      ArrayList<Double> bills = new ArrayList<>();

      for (int i=0; i<12; i++)
      {
         bills.add(29.99);
      }

      for (int i=0; i<12; i++)
      {
         double x = bills.get(i);
         System.out.println(" Month " + (i+1) + ": $" + x);
      }
   }
}
```

ArrayList: *Try it yourself*

```
import java.util.ArrayList;

public class Prog25_02
{
   public static void main(String[] args)
   {
      ArrayList<Double> bills = new ArrayList<>();

      for (int i=0; i<12; i++)
      {
         bills.add(29.99);
      }

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

Arrays: *Try it yourself*

```
import java.util.ArrayList;
import javax.swing.*;

public class Prog25_03
{
    public static void main (String args [ ])
    {
        ArrayList<Double> a =
                new ArrayList<>();

        // Fill the array with powers of two
        for ( int i = 0; i < 5; i++)
        {
                a.add(Math.pow(2, i));
        }

        // Prepare for displaying the values of a
        String s = "";

        for ( int i = 0; i < a.size(); i++)
        {
                s = s + Math.round(a.get(i)) + "\n";
        }

        JOptionPane.showMessageDialog ( null, s, "Output",
                        JOptionPane.INFORMATION_MESSAGE);

    }
}
```

***Size* of ArrayList a**

PRACTICE

Program 25_04:

Write a Java program that stores 10 random integers (each in [10, 30]) in an ArrayList variable and

- finds and prints the average
- finds and prints the number of values less than the average
- finds and prints the number of values greater than the average

ArrayList: *Try it yourself*

```
import java.util.*;
public class Prog25_05
{
   public static void main(String args[])
   {
      Random rnd = new Random();

      ArrayList<Integer> a = new ArrayList<>();

      for (int i = 0; i < 10; i++)
      {
         a.add(rnd.nextInt(10));
      }

      System.out.println(a);

      if (a.contains(0)) System.out.println("Found!");
      else  System.out.println("Not found!");
   }
}
```

ArrayList: *Try it yourself*

```
import java.util.ArrayList;

public class Prog25_06 {
    public static void main(String args[])
    {
        ArrayList<String> list = new ArrayList<>();

        list.add("Mango");
        list.add("Orange");
        list.add("Banana");
        list.add("Orange");
        list.add("Kiwi");
        list.add("Orange");
        list.add("Apple");

        System.out.println(list);
        list.remove("Orange");
        System.out.println(list);
    }
}
```

PRACTICE

Program 25_07:

Modify Program 25_06 to remove all occurrences of "Orange".

PRACTICE

Program 25_08:

Write a Java program that populates an ArrayList variable with 10 random integers, asks the user to enter a number, and prints the location of the first occurrence of the number in the ArrayList, if found.

Note: use Java documentation to find an utility that determines location of a value in an ArrayList.