CSE 240
Homework 1
Programming with Java

1. What This Assignment Is About?

- Classes (methods and attributes)
- Objects
- Arrays of Primitive Values
- Arrays of Objects
- Recursion
- for and if Statements
- Selection Sort

2. Use the following Guidelines

- Give identifiers semantic meaning and make them easy to read (examples `numStudents`, `grossPay`, etc.)
- Use upper case for constants.
- Use title case (first letter is upper case) for classes.
- Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with if, switch and loop statements. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

For each file (class) in your assignment, provide a heading (in comments) which includes:

- Author (your name).
- A description of what this program is doing.

3. Part 1. Primitive Types, Searching, Recursion (32 points)

a) Create a class **`Homework`** (in a file Homework.java)

b) Create a **`static`** method **`initializeArray`** that receives as a parameter an array of integers. Use a for loop and an if statement to put **'a'** at the odd indexes of the array and **'b'** at the even indexes.

c) Create a **`static`** method **`printArray`** that receives as a parameter an array of

characters. Use a for statements to print all the elements in the array.

d) Create a **`static`** method **`selectionSort`** that receives as a parameter an array of characters and order its elements in **descending order**. Implement Selection Sort algorithm. It should be Selection Sort, not Bubble Sort, not Quick Sort, etc. If you do not remember selection sort, this link could be useful: SelectionSort

e) Create a **`static`** **recursive** method named **`factorial`** that calculate and returns the factorial of a number. The method receives a number (integer number) as parameter.

f) Copy the main method in your class, again the **`main`** method (MainMethodForHomework.txt in the assignment description in canvas):

4. Grading Criteria for the part 1

If the program does not compile the grade is 0 (zero)

2.0 points: file contains header information
2.0 points: adequate comment to explain every method
2.0 points: consistent indentation and spacing
2.0 points: naming conventions are applied
6.0 points: `initializeArray` method
6.0 points: `printArray` method
6.0 points: `selectionSort` method
6.0 points: `factorial` method

5. Part 2 Classes, Objects, and Arrays (68 points)

In this assignment, we will be making a program that create an examination seating with a number of rows and columns specified by a user.

Use the file **`HomeworkTwo.java`** (attached in the assignment description in canvas). **Do not change the content of this file.**

Save **`HomeworkTwo.java`** and all files requested below in the same folder.

Step 1

First, create a file named **`Student.java`** and define a class **`Student`**. It should have two instance variables, **`firstName`** (String) and **`lastName`** (String). The class **must** include the following methods:

Method	Description of the Method
public Student( )	Constructs a Student object by assigning the default string

	"foo" to lastName and “bar” to firstName.
public Student(String info)	Constructs a Student object using the string info. Use the split method of the String class to extract first name and last name, then assign them to each instance variable of the Student class. An example of the input string is: `John/Doe`
public String getLastName( )	It should return the instance variable lastName.
public String getFirstName( )	It should return the instance variable firstName.
public String toString( )	It should return a string containing the initial character of the first name, a period, the initial character of the last name, and a period. This does not apply to bar and foo. An example of such string for the student John Doe is: `J.D.`

Step 2

Second, create a class called **`Classroom`**. This class should be defined in a file named **`Classroom.java`**. The class contains a 2-dimensional array (called **`arrangement`**) of Student objects. (If you do not remember multidimensional array, this link could be useful: Multi-Dimensional Array) The class must include the following methods:

Method	Description of the Method
Public Classroom (int rowNum, int columnNum)	It instantiates a two-dimensional array of the size "rowNum" by "columnNum". Then it initializes each student element of this array using the constructor of the class Student without any parameter. So, each student will have default values for its instance variables.
private Student getStudentAt (int row, int col)	.It returns the student object at the indexes row and col (specified by the parameters) from the 2D array "`arrangement`".
public boolean setStudentAt (int row, int col, Student data)	The method assign the Student object referenced as "data" to the seat at "row" and "col". If the seat has a default student, i.e., a student with the last name "foo" and the first name "bar", then

	we can assign the new student "data" to that seat and the method returns true. Otherwise, this seat is considered to be taken by someone else, the method does not assign the student and returns false.
public boolean isValid (int row, int col)	The method checks if the parameters row and col are valid. If at least one of the parameters "row" or "col" is less than 0 or larger than the last index of the array then it returns false. Otherwise it returns true.
public String toString( )	Returns a String containing the information of the array "arrangement". It should use the toString method of the class Student and return the following format: **The current seating:** **D.J. bar.foo. E.T.** **bar.foo. bar.foo. S.W.** **T.C. A.T. bar.foo.**

After compiling **`Student.java`**, **`Classroom.java`**, and **`HomeworkTwo.java`** files then execute **`HomeworkTwo.class`**.

Sample Output: (the inputs entered by a user are shown in bold)

Make sure that your program works **at least** with this scenario.

`C:\MyJava\applications>java HomeworkTwo`

```
How many rows do you want?
3
How many columns do you want?
3
Capture a student information (name/lastname) or enter "Q" to quit.
Mickey/Mouse
Capture the row number where the student wants to sit:
1
Capture the column number where the student wants to sit:
2
The seat at row 1 and column 2 is assigned to M.M.

The current seating:
bar.foo. bar.foo. bar.foo.
bar.foo. bar.foo. M.M.
bar.foo. bar.foo. bar.foo.

Capture a student information (name/lastname) or enter "Q" to quit.
```

```
Daisy/Duck
Capture the row number where the student wants to sit:
2
Capture the column number where the student wants to sit:
0
The seat at row 2 and column 0 is assigned to D.D.

The current seating:
bar.foo. bar.foo. bar.foo.
bar.foo. bar.foo. M.M.
D.D. bar.foo. bar.foo.

Capture a student information (name/lastname) or enter "Q" to quit.
Clarabelle/Cow
Capture the row number where the student wants to sit:
2
Capture the column number where the student wants to sit:
1
The seat at row 2 and column 1 is assigned to C.C.

The current seating:
bar.foo. bar.foo. bar.foo.
bar.foo. bar.foo. M.M.
D.D. C.C. bar.foo.

Capture a student information (name/lastname) or enter "Q" to quit.
Max/Goof
Capture the row number where the student wants to sit:
0
Capture the column number where the student wants to sit:
0
The seat at row 0 and column 0 is assigned to M.G.


The current seating:
M.G. bar.foo. bar.foo.
bar.foo. bar.foo. M.M.
D.D. C.C. bar.foo.

Capture a student information (name/lastname) or enter "Q" to quit.
Horace/Horsecollar
Capture the row number where the student wants to sit:
5
Capture the column number where the student wants to sit:
1
row or column number is not valid.
A student Horace Horsecollar is not assigned a seat.

Capture a student information (name/lastname) or enter "Q" to quit.
Sylvester/Shyster
Capture the row number where the student wants to sit:
2
Capture the column number where the student wants to sit:
0
The seat at row 2 and column 0 is taken.

Capture a student information (name/lastname) or enter "Q" to quit.
Snow/White
Capture the row number where the student wants to sit:
-1
Capture the column number where the student wants to sit:
0
```

```
row or column number is not valid.
A student Snow White is not assigned a seat.

Capture a student information (name/lastname) or enter "Q" to quit.
Jiminy/Criket
Capture the row number where the student wants to sit:
0
Capture the column number where the student wants to sit:
2

The seat at row 0 and column 2 is assigned to J.C.

The current seating:
M.G. bar.foo. J.C.
bar.foo. bar.foo. M.M.
D.D. C.C. bar.foo.

Capture a student information (name/lastname) or enter "Q" to quit.
Q
```

Grading Criteria for the part 2

If the program does not compile the grade is 0 (zero)

4.0 points: Every file contains header information
4.0 points: adequate comment to explain every method
4.0 points: consistent indentation and spacing
4.0 points: naming conventions are applied
6.5 points: The two constructors of Student class are correct
6.5 points: Accessor methods for lastName and firstName of Student are correct
6.5 points: The toString method in Student class is correct
6.5 points: The Constructor for the class Classroom is correct
6.5 points: getStudentAt method is correct
6.5 points: setStudentAt method is correct
6.5 points: isValid method is correct
6.5 points: The toString method in Classroom class is correct