

CSE 240

Homework 4 – Programming with C++

1. What This Assignment Is About:

- Classes and Objects
- Methods
- 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 ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.
- You should **only submit the following files**:

• homework.h	• homework.cpp
• patron.h	• patron.cpp
• theatre.h	• theatre.cpp
• main_part1.cpp	• main_part2.cpp

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

- **The assignment number.**
- **Its author (your name).**
- **A description of what this program is doing.**

YOU ARE NOT ALLOWED TO MAKE ANY CHANGES TO THE MAIN FUNCTION OTHER THAN HEADERS.

3. Part 1. Primitive Types, Searching, Recursion (35 points).

- a) Create a class Homework (in a file **homework.h** and **homework.cpp**). Review the lectures and textbook to learn about the content of each file.
- b) Create a function **initializingArray** that receives two parameters: an array of integers and the array size. Use a for loop and an if statement to put 0s in the odd indexes of the array and 1s in the even indexes. **(Use pointers to pass an array of integers as parameter)**

- c) Create a function **printingArray** that receives as parameters an array of integers and the array size. Use a for statements to print all the elements in the array. (**Use pointers to pass an array of integers as parameter**)
- d) Create a function **selectionSort** that receives as parameters an array of integers and the array size and order the array element in ascending 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:
<https://www.geeksforgeeks.org/selection-sort/>
- e) Create a recursive function that calculate and returns the **factorial** of a number. The function receives the number (integer number) as parameter
- f) Create a file **main_part1.cpp**. Copy the main function for part 1 from the canvas assignment description in your class.

Grading Criteria for the part 1

05 pts: file contains header information and adequate comment to explain every function
 05 pts: consistent indentation and spacing
 02 pts: 3 files (1 file *.h, 2 files *.cpp)
 05 pts: selectionSort
 05 pts: printingArray
 05 pts: initializingArray
 05 pts: factorial

4. Part 2 Structs and Arrays (65 points).

In this assignment, we will be making a program that reads in patrons' information and create a movie theatre seating with a number of rows and columns specified by a user. Then it will attempt to assign each patron to a seat in a theatre.

Create a file **main_part2.cpp**. Copy the main function for part 2 from the canvas assignment description in your class.

[Include all the following requested code in new *.cpp and *.h files, as needed.](#)

Step 1.

First, you need to create a class **Patron**. Create patron.cpp and patron.h files. It should contain two variables, lastName (char [25]) and firstName (char [25]). Both should be private. In addition, the following functions should be defined. All of them are public

Method	Description of the Method
Patron ()	Constructs a Patron object by assigning the default string " ### " to both instance variables, lastName and firstName.
Patron	Constructs a Patron object using the string containing patron's info. Use

(char* patronInfo)	the strtok function to extract first name and last name, then assign them to each instance variable of the Patron class. An example of the input string is: David/Johnson
char* getLastName ()	It should return the instance variable lastName.
char* getFirstName ()	It should return the instance variable firstName.
char* toString ()	It should constructor a string containing the initial character of the first name, a period, the initial character of the last name, and a period, then it returns it. An example of such string for the patron David Johnson is: <i>D.J.</i>

Step 2.

You will be creating a class called **Theatre**. Create theatre.cpp and theatre.h files. The class TheatreSeating will contain a 2-dimensional array called "arrangement" of Patron objects at its instance variable. The class Theatre **must** include the following constructor and methods. You will also need to create variables to check rows and columns for the seating (If your class does not contain any of the following methods, points will be deducted.)

Method	Description of the Method
Theatre (int rowNum, int columnNum)	It instantiates a two-dimensional array of the size "rowNum" by "columnNum" specified by the parameters. Then it initializes each patron element of this array using the constructor of the class Patron without any parameter. So, each patron will have default values for its instance variables.
Patron* getPatronAt (int row, int col)	It returns a patron at the indexes row and col (specified by the parameters of this method) of the array "arrangement".
bool assignPatronAt (int row, int col, Patron *tempPatron)	The method attempts to assign the "tempPatron" to the seat at "row" and "col" (specified by the parameters of this method). If the seat has a default patron, i.e., a patron with the last name "###" and the first name "###", then we can assign the new patron "tempPatron" 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 patron and returns false.

bool checkBoundaries (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 (note that the number of rows and columns can be different), then it returns false. Otherwise it returns true.
char* toString()	Returns a String containing information of the "arrangement". It should show the list of patrons assigned to the arrangement using the toString method of the class Patron (it shows initials of each patron) and the following format: The current seating ----- D.J. ##. E.T. ##. ##. S.W. T.C. A.T. ##. Please see the sample output listed below.

After compiling all files, you need to execute it.

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

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

```
Please enter a number of rows for a theatre seating.
3
Please enter a number of columns for a theatre seating.
3
Please enter a patron information or enter "Q" to quit.
Mickey/Mouse
A patron information is read.
Mickey/Mouse
Please enter a row number where the patron wants to sit.
1
Please enter a column number where the patron wants to sit.
2
The seat at row 1 and column 2 is assigned to the patron M.M.

The current arrangement
-----
##. ##. ##.
##. ##. M.M.
##. ##. ##.

Please enter a patron information or enter "Q" to quit.
Daisy/Duck
A patron information is read.
Daisy/Duck
Please enter a row number where the patron wants to sit.
2
```

Please enter a column number where the patron wants to sit.

0

The seat at row 2 and column 0 is assigned to the patron D.D.

The current arrangement

##. ##. ##.

##. ##. M.M.

D.D. ##. ##.

Please enter a patron information or enter "Q" to quit.

Clarabelle/Cow

A patron information is read.

Clarabelle/Cow

Please enter a row number where the patron wants to sit.

2

Please enter a column number where the patron wants to sit.

1

The seat at row 2 and column 1 is assigned to the patron C.C.

The current arrangement

##. ##. ##.

##. ##. M.M.

D.D. C.C. ##.

Please enter a patron information or enter "Q" to quit.

Max/Goof

A patron information is read.

Max/Goof

Please enter a row number where the patron wants to sit.

0

Please enter a column number where the patron wants to sit.

0

The seat at row 0 and column 0 is assigned to the patron M.G.

The current arrangement

M.G. ##. ##.

##. ##. M.M.

D.D. C.C. ##.

Please enter a patron information or enter "Q" to quit.

Horace/Horsecollar

A patron information is read.

Horace/Horsecollar

Please enter a row number where the patron wants to sit.

5

Please enter a column number where the patron wants to sit.

1

row or column number is not valid.

A patron Horace Horsecollar is not assigned a seat.

Please enter a patron information or enter "Q" to quit.

Sylvester/Shyster

A patron information is read.

Sylvester/Shyster

```

Please enter a row number where the patron wants to sit.
2
Please enter a column number where the patron wants to sit.
0
The seat at row 2 and column 0 is taken.
Please enter a patron information or enter "Q" to quit.
Snow/White

A patron information is read.
Snow/White
Please enter a row number where the patron wants to sit.
-1
Please enter a column number where the patron wants to sit.
0

row or column number is not valid.
A patron Snow White is not assigned a seat.
Please enter a patron information or enter "Q" to quit.
Jiminy/Criket

A patron information is read.
Jiminy/Criket
Please enter a row number where the patron wants to sit.
0
Please enter a column number where the patron wants to sit.
2

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

The current arrangement
-----
M.G. #.#. J.C.[1]SEP
#.#. #.#. M.M.
D.D. C.C. #.#.

Please enter a patron information or enter "Q" to quit.
Q

```

Grading Criteria for the part 2

- 05 pts: Every file contains header information
- 05 pts: adequate comment to explain every method
- 05 pts: consistent indentation and spacing
- 05 pts: it compiles
- 03 pts: 5 files (2 file *.h, 3 files *.cpp)
- 05 pts: Two constructors of Patron are correct
- 05 pts: Accessor methods for lastName and firstName of Patron are correct
- 05 pts: toString method of Patron is correct
- 05 pts: Constructor, Theatre(int,int) is correct
- 05 pts: getPatronAt(int,int) method is correct
- 10 pts: assignPatronAt(int,int,Patron) method is correct
- 05 pts: checkBoundaries(int,int) method is correct
- 05 pts: toString method is correct