

## Calendar Assignment- Part 3

### Simple Scheduling

CS 140

#### Topics:

File I/o (input / output), Arrays

#### Learning Outcomes:

- Practice Reading in files to java and processing file text content
- Become comfortable with using arrays to store and retrieve data
- Design and develop programs in accepted style
- Develop further the ability to enhance a program to increase functionality
- Execute the proper steps to write out desired content to a file

Once finished with this part of the assignment, experience with the content up through chapter 7 should be obtained.

#### Task:

For the third part of the assignment, a few more pieces of functionality will be added. Each piece added will add a feature that applies one of the subjects covered lately in class. One such feature will be the ability for the calendar to store events for printing in the calendar squares. This feature will implement arrays. Another feature added will be functionality to read in events from a file and insert them into the events array. Finally, the calendar will be expanded to include the ability to print a month calendar draw to a file. This will use file output.

#### Task one: - Event Planning

This task will involve adding a menu item to the menu of the calendar. When the command "ev" is entered, a new action should be started. The event planning action should prompt the user for an event. The event should be entered in the form of "MM/DD event\_title". After parsing the event, should be stored in a global array that will contain all events planned for that year. Event array should be a multidimensional array. It should be size 12(number of months in a year), with each sub array being large enough hold a single event for every day for the month. Each sub array, should be **exactly** the correct size for the number of days in that given month. For example, eventArray[11].length == 31, where 11 is the month December.

Once the event is parsed and stored, if the month calendar is drawn that contains scheduled events, those events should be presented in the calendar. If there is an event in a

day, the title of the event should be placed within the blank space within the square of the day.

### **Task two: - File Reading**

Now that event planning is in place, if an event file exists, events should be read into the calendar when the calendar is first loaded. When the calendar starts, it will look for a file by the name "calendarEvents.txt". If that file is in the same directory as the program, the calendar will read in the events in the file. Given the event that the event file does not exist, no events will be read into the events array. Either the given example events file or an events file by the same name of your creation can be used to populate the calendar events. The events in the file must be of the form as entered events from the keyboard ("MM/DD event\_title"). Events from the file will be read into the same events array from the first task. Once the calendar is drawn, events for the month should be placed within the appropriate dates squares, just like in the first task.

### **Task three: - File printing**

For this task, the calendar will be printing the same thing that it might to the screen but this time it will be printing to a file. This includes the ascii art for a month. Add a new command "fp" to the menu. Once this command has been entered, the user will be prompted to enter a month to print. After the month to print has been obtained, the program should proceed to ask the user for the name of a file to which it will print the calendar. The program will next proceed to print the calendar with the appropriate events into the file. Make sure the file is closed once writing is completed.

### **Style:**

It is important that you get used to writing code in good style. What is demonstrated in examples in class is considered good style. Additionally, you should look at the style guide located on Canvas. Badly styled code will lose points.

### **Other Information:**

The tasks laid out in this document should be completed to the best of your ability. There is less guidance as to where to implement what for a few reasons. As the last assignment, it should be slightly more challenging. Secondly this assignment has been developing over the quarter so you should know where things need to go well.

You may not use material beyond Chapter 7 for this assignment. You may make use of the tools we have learned up through Chapter 7 and those stated in the Helpful Information section on this specification. String, Math, Scanner and other objects and their functionality that we have gone over in class and that are covered in the book through

Chapter 7 are acceptable to use. Please ask your teacher before implementing additional functionality found within java you may run into online.

### **Extra Credit Opportunities (Optional):**

You can choose to do as many or as few of these as you would like. Each one completed successfully and in good style will earn some extra credit points towards this assignment.

1. Some extra credit will be given to those who build their program with the ability to read events from a filename that the user gives the program. Thus, if the user wishes to read files from chaos.txt, the program reads in events from that file. When this feature is complete, Exceptions that may be thrown from errors with non-existent files must be handled.
2. Another opportunity for extra credit will be for a custom function added to the Calendar. This will include adding the command to the allowed commands list as well as functionality to support it. The amount of extra credit for this part will be determined by the significance of the feature added.

### **Grading:**

Style	10
Event Planning	30
Reading events from file	20
Printing calendar to file	20
Overall program functionality	20
<b>Total:</b>	<b>100</b>