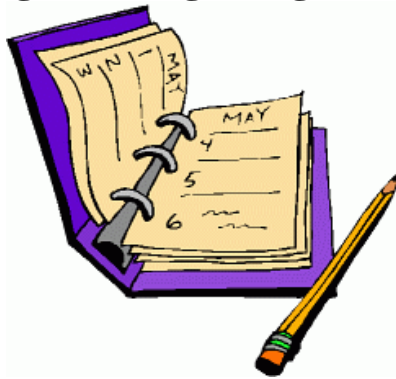


Programming Assignment 6



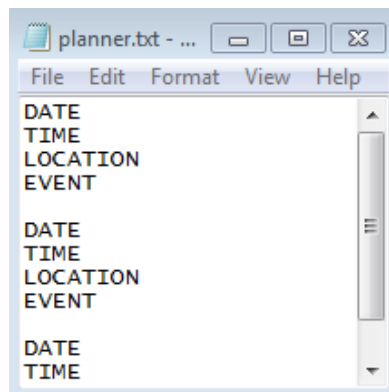
<http://www.clipartkid.com/images/443/student-planner-clipart-planners-fezE3V-clipart.gif>

Objectives:

- Practice reading from files
- Practice writing to files
- **NO ARRAYS!**

Assignment:

In this assignment, you are going to write a program that simulates a planner. The program allows the user to add a new event to the planner, display all the events currently stored in the planner, or to quit the program. Since this is a planner, the data needs to be saved to a file called **planner.txt** so that the events in the planner aren't lost each time the program ends. In the file, each event will be written in the same format. Your file **MUST** be in this format.



If the user chooses to add an event, ask them to enter the date, time, location, and name of the event. Then, write them to the information to the file in the correct order. If the user chooses to display the events in the planner, print out all the events in the file, in the order they appear in the file. If there are no events in the planner, print a message saying no events have been added. If the user chooses to quit, the program ends.

Using a temporary file to store the events will be useful when adding an event.

Sample Execution 1:

Personal Planner
=====

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 2

No events added!

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 1

Enter date (MM/DD/YY): 03/23/17

Enter time (HH:MM): 14:00

Enter location: My Office

Enter event: Office Hours

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 2

03/23/17

14:00

My Office

Office Hours

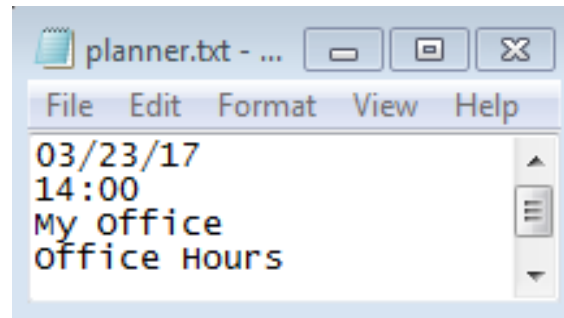
Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 3

planner.txt after sample execution 1:



Sample Execution 2:

Personal Planner

=====

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 2

03/23/17
14:00
My Office
Office Hours

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 1

Enter date (MM/DD/YY): 03/30/17

Enter time (HH:MM): 09:30

Enter location: HMPH 308

Enter event: Exam 3

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 2

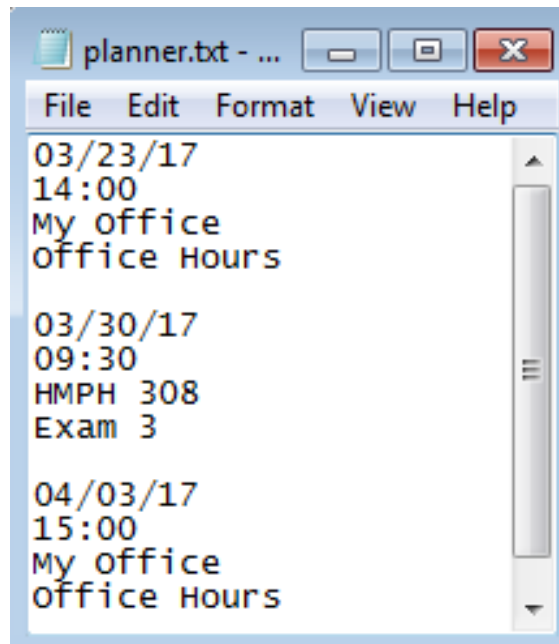
03/23/17
14:00
My Office
Office Hours

03/30/17
09:30
HMPH 308
Exam 3

Options:

=====

planner.txt after sample execution 2:



1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 1

Enter date (MM/DD/YY): 04/03/17

Enter time (HH:MM): 15:00

Enter location: My Office

Enter event: Office Hours

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 2

03/23/17

14:00

My Office

Office Hours

03/30/17

09:30

HMPH 308

Exam 3

04/03/17

15:00

My Office

Office Hours

Options:

=====

1. Add Event
2. Display Events
3. Quit

Enter options (1, 2, or 3): 3

Requirements:

- Use an updated comment block
- Your program should use the following comment block at the very beginning of your program.

```
// Name: Your Name                      Date Assigned: Fill in
//
// Course: CSCI 2003 60357              Date Due: Fill in
//
// Instructor: Ms. Greer
//
// File name: Fill in
//
// Program Description: Brief description of what the program does.
```

- Use appropriate comments throughout the program
- Make good use of whitespace
- Your output should look exactly like the sample output if using the same data.

Deliverables:

- Planner.java file
- Upload 1 file to Moodle

Grading:

Total Points	15 points
Planner class	15 points
Displays menu correctly	1 point
Gets option from user correctly	1 point
Adds Event Correctly	
Happens when user enters 1	1 point
Stores previous data from planner.txt correctly (to avoid losing previous events)	1 point
Gets date, time, location, and event correctly	1 point
Opens planner.txt for writing correctly.	1 point
Writes info to file in correct format.	1 point
Close files correctly.	1 point
Displays all events in planner.txt correctly.	
Happens when user enters 2	1 point
Opens file for reading	1 point
Uses loop to read from file	1 point
Displays data in planner.txt correctly	1 point
Closes file	1 point
Quits program correctly	
Program ends when user enters 3	1 point
Doesn't display menu again	1 point
Not enough comments/whitespace	-1 point
Output does not match the sample executions given in the assignment	-1 point
Bad variable names, method names, and/or class names	-1 point