

Assignment 6 – Season Ticket Kiosk

Maximum Points = 50

The purpose of this lab is to introduce you to loops and continue the use of the Scanner, File classes and conditional (decision) statements in the Java programming language. Your program will simulate the user interface for a kiosk used for selling season tickets.

As a new student intern at the CHattahoochee valley Arts and Origami Society (CHAOS), you have been asked to write the software for the newly released SEason Ticket Kiosk (SETIKI) from NCR. Your program will read data from a text file and then provide the user with a menu of choices to either list the season's performances or purchase tickets. The SETIKI allows tickets to be purchased with credit card, check, or cash.

This assignment builds on Assignment 4 and 5, but does NOT use the Applet class.

- ❖ Your program must read the data from a file called schedule.txt
- ❖ The data in the file will consist of 10 performances (each performance will consist of a Type of performance (T or M), Title/Name, Date, time, location, and cost) where each component will be separated by the hash (#) symbol as shown below:

```
T#Hamlet#October 17, 2011#7:30 PM#Riverpark Theater, CSU#12.50
M#Loretta Lynn#November 22, 2011#7:30 PM#RiverCenter#25.25
:
```

- ❖ Your program must include a class that will hold the performance data (complete with constructor(s), get and set methods for each instance variable, and a toString method that returns the contents of the instance variables).
- ❖ Your GUI should display an appropriate greeting to to the SETIKI user and then
- ❖ ***provide the user with a menu that allows the user to a) view only musical performances, b) view only theatrical performances, c) view both types of performances, d) purchase only musical performances tickets, e) purchase only theatrical performances tickets, or f) purchase both types of performances tickets, or end all transactions. [Note that a user may want to do more than one transaction during their session.] {TEST to make sure that the user selects a correct transaction}***
- ❖ If the user wishes to view the list of performances, the SETIKI needs to display the schedule listing the ***selected*** performances. ***[Note that there may be between 0 and 10 performances selected.] {provide an appropriate comment if there are no (0) performances to display}***
- ❖ ***If the user is ready to purchase tickets, ask them for the type of payment. Provide a 10% discount for those selecting ALL (both musical and theatrical performances).***
 - ***For credit cards and checks, ask them to enter the card or check and “print” a receipt.***
 - ***For cash, “read” in the amount of cash tendered, and “print” a receipt and the change [you may use the CashRegisterSimulator class on page 156-157, adding up to \$100 bills].***

HINT: Reuse as much code as possible from Assignment 5.

EXTRA CHALLENGES:

- Replace the CashRegisterSimulator with an ENHANCED CashRegisterSimulator that directs the cashier how to give change (P4.11 on pages 166-167.)

Make sure that your program uses proper indentation and complete documentation. See <http://csc.columbusstate.edu/summers/NOTES/1301/style.htm> for guidelines.

The program heading should occur at the top of the program and should include:

```

/**
 * PROGRAM SPECIFICATIONS
 * NARRATIVE DESCRIPTION:
 *
 * @author (your name)
 * @version (date)
 */

```

(Due before 8 a.m. on Wednesday, **November 9, 2011**) Submit a .doc file containing the UML class diagram showing inheritance for all the classes used in your program and your timesheet documenting your time. [10 pts]

(Due before 8 a.m. on Wednesday, **November 16, 2011**) Submit your .java files containing your program and your timesheet documenting your time to the dropbox in WebCT..

Grades are determined using the following scale:

- Runs correctly.....:___/10
- Correct output.....:___/10
- Design of output.....:___/8
- Design of logic.....:___/10
- Standards.....:___/7
- Documentation.....:___/5

[Grading Rubric](#) ([Word document](#))