CS1073 Assignment #5

Fall 2012

Due: Friday, November 9 by NOON.

Please note the following:

- **As usual, a paper copy must be placed in the appropriate assignment bin on E level of Head Hall (where Head Hall and Gillin Hall connect).**
- **In addition, your files must be uploaded to Desire2Learn (please follow the instructions below).**

The purpose of this assignment is to allow you to gain practice with GUIs and event-driven programming. It also gives you a chance to review some of the earlier course topics.

For this assignment, please create all of your GUIs as applications (not applets).

---

1. Creating a Graphical GPA Calculator

a. The University of New Brunswick uses a letter grade system. At the end of each course, each student is given a letter grade that represents his/her performance in that course. Each letter grade corresponds to equivalent grade points. These equivalences are shown in the following table (from the UNB Undergraduate Calendar):

Letter Grade	Grade Points
A+	4.3
A	4.0
A-	3.7
B+	3.3
B	3.0
B-	2.7
C+	2.3
C	2.0
D	1.0
F	0.0
WF	0.0

UNB also follows a credit hour system. This means that each course has a certain number of credit hours associated with it. For example, CS1073 is a 4 credit hour course. Courses with a higher credit hour value have a larger impact on a student's Grade Point Average (GPA) than those with fewer credit hours.

To calculate the total points earned in a single course, we simply take the grade point value and

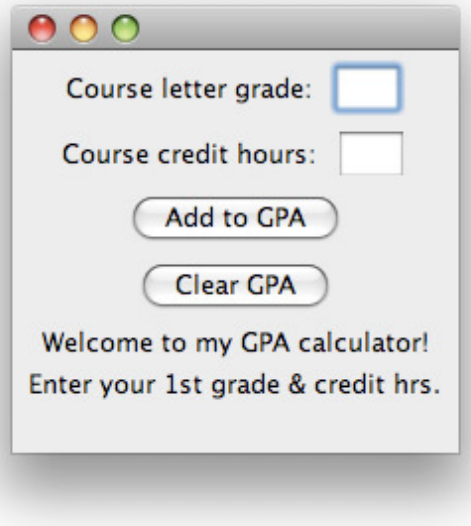
multiply it by the number of credit hours. That means that if a student receives a grade of B in CS1073, he/she earns 12.0 points (3.0 grade points * 4.0 credit hours).

To calculate a student's Cumulative GPA, we add up all the points that he/she has earned in all completed courses, and we divide by the sum of all the credit hours for those courses. Consider the following example:

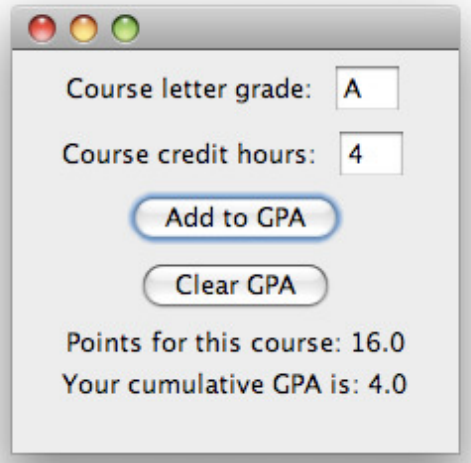
```
Fall 2012 term for John Doe:
CS 1073   (4 ch) A+  ==>  17.2 pts
CS 1203   (3 ch) A   ==>  12.0 pts
GER 1001  (3 ch) B+  ==>   9.9 pts
PHYS 1061 (3 ch) B   ==>   9.0 pts
PHYS 1091 (2 ch) A   ==>   8.0 pts


John's GPA: (17.2 + 12.0 + 9.9 + 9.0 + 8.0) / (4 + 3 + 3 + 3 + 2) = 3.74
```

Create a graphical GPA calculator application that allows users to enter the letter grade and the number of credit hours for a course, and then request his/her GPA. The GUI should display the resulting points for that course and also display the student's cumulative GPA. The user should then be able to enter the results for another course, and ask the application to recalculate the cumulative GPA. At any time, the user should be able to clear the cumulative GPA, and start new calculations.

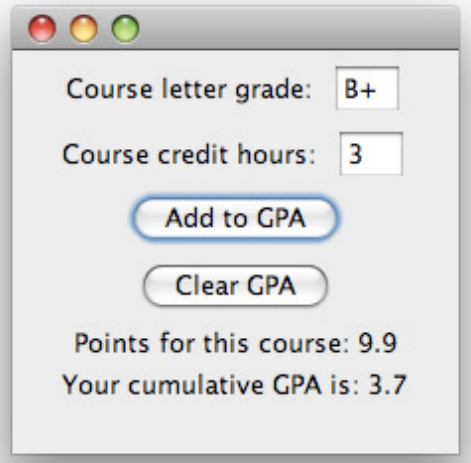
Please follow these sample screen shots:



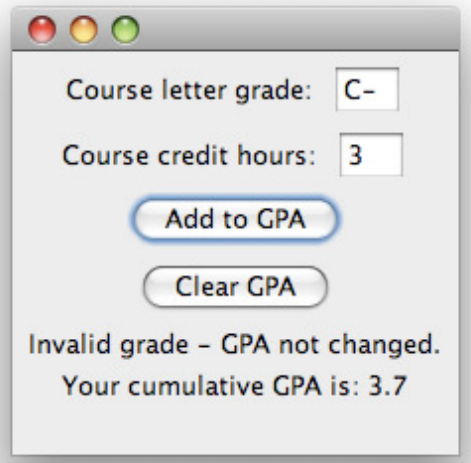
(GUI appearance when first loaded - no user interction yet).



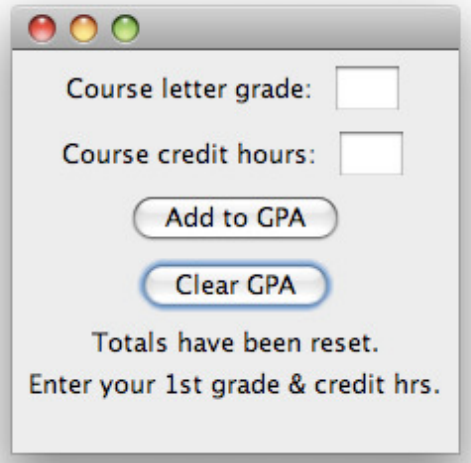
(User enters data for the first course and presses the "Add to GPA" button).



(User enters data for another course and presses the "Add to GPA" button).



(User enters data for the another course, but with an invalid grade, and presses the "Add to GPA" button).



(User presses the "Clear GPA" button).

Note: When displaying the resulting points and cumulative GPA, these values should be displayed with one decimal digit (hint: use the NumberFormat class).

For this question, you may simply use FlowLayout to arrange the components within the content pane. Choose a reasonable starting width and height so that everything is visible (i.e. nothing is cut off) when the GUI is first displayed.

b. Test your program with the sample data listed above for John Doe. Test your program with an invalid grade entry as well. Also, test the "Clear GPA" button. Once you are sure that the application is working correctly, print your code. In addition, print 4 screenshots: one screenshot before any calculations have been performed, one after entering John Doe's information, one after an invalid grade entry, and one after pressing the "Clear GPA" button.

---

2. Creating a Graphical Cash Register for a Movie Theatre

Intro: A local independent movie house has just opened in Fredericton. In addition to selling movie tickets, they will also be selling some concessions (popcorn, pop and candy). They have asked you to create a graphical application that they can run on their laptop to calculate the total ticket and food sales for each customer.

Aside: To help drum up business for this new movie house, the owner has distributed various different coupons (through the local newspaper, the web, etc.) These coupons each have a specific value (e.g. $1.50) and a customer can use these coupons towards the purchase of movie tickets and/or concessions (i.e. they are treated the same as cash at this cinema).

Note: There are 6 image files that you should use in your solution to this question. You can download them here: as5-images.zip (Download this file and unzip it. Place all of these image files in the folder where you will create your .java files for this question; do not place them in a subfolder.)

a. First, write a `ProductButton` class. I have already started this class for you; just follow this link: ProductButton.java (You do not need to add any new instance variables or methods; you only need to complete the constructor method. Read through the javadoc comments carefully and make sure that the constructor works as described.)

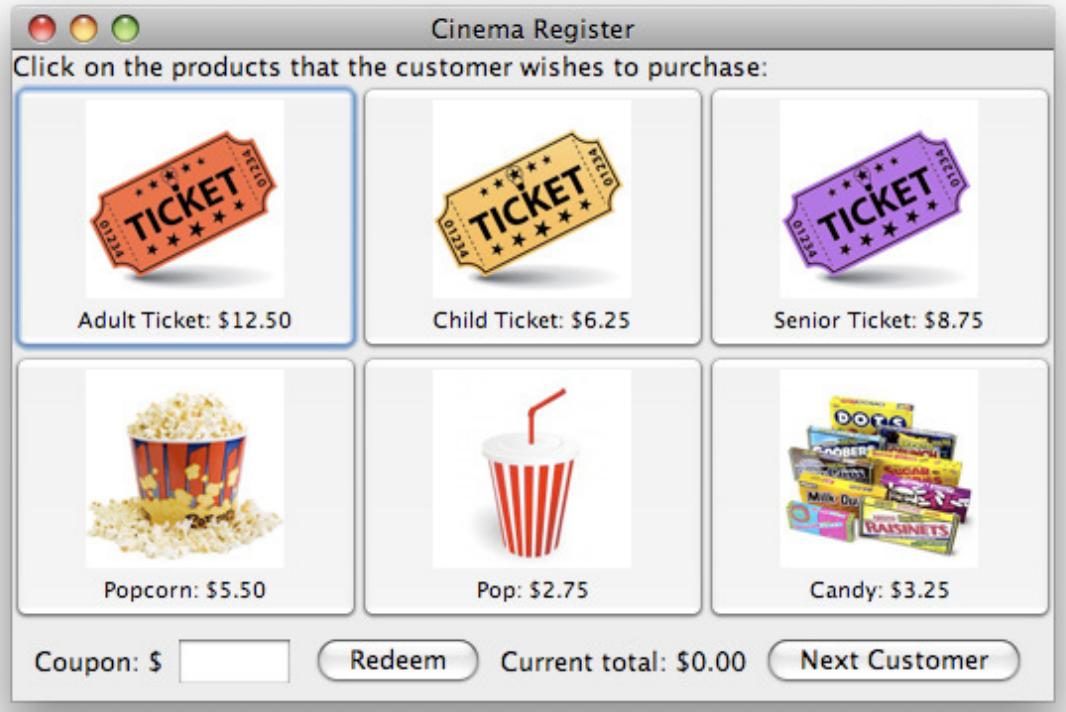
b. Next, test your class from part a by creating a test driver. Your test driver should be in a new file. In your test driver, create a frame with one ProductButton in it. You may use any of the images that you downloaded at the start of this question.

   Once you have seen that the ProductButton class is working correctly, you may move on to part c.

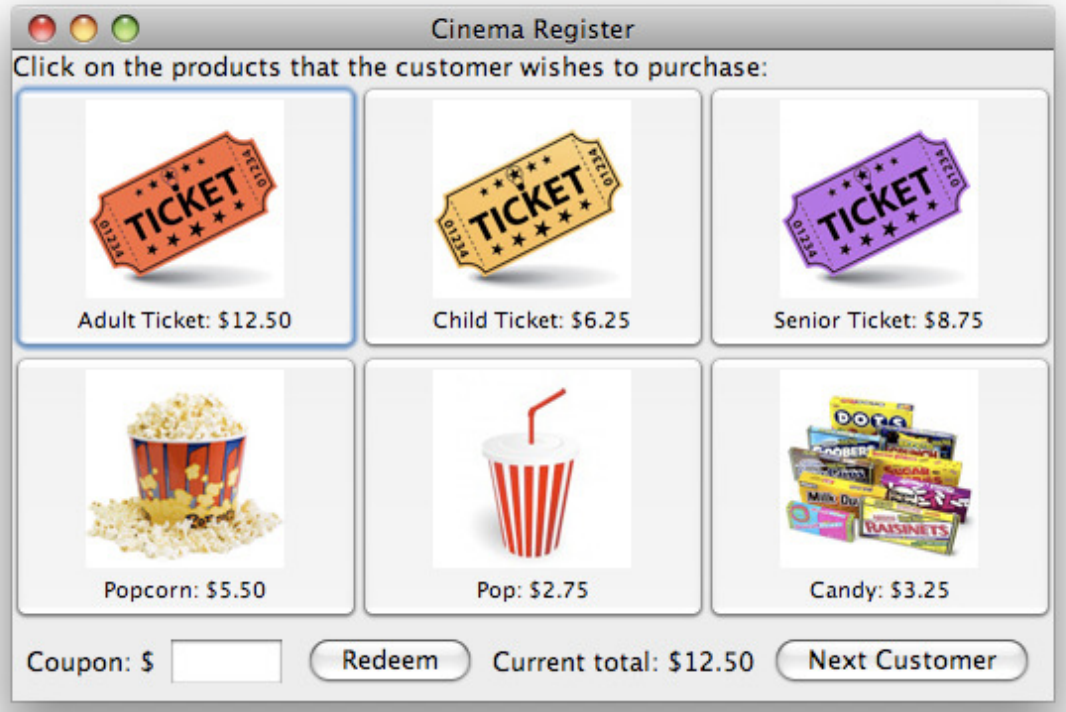
   Note: The code for part b does not need to be handed in for marking. The purpose of this part is to ensure that your ProductButton class is functioning properly before you attempt part c.

c. Now, in a new class, named `CinemaFrame`, you will create a graphical cash register application. **Your solution must make use of the ProductButton class from part a.**

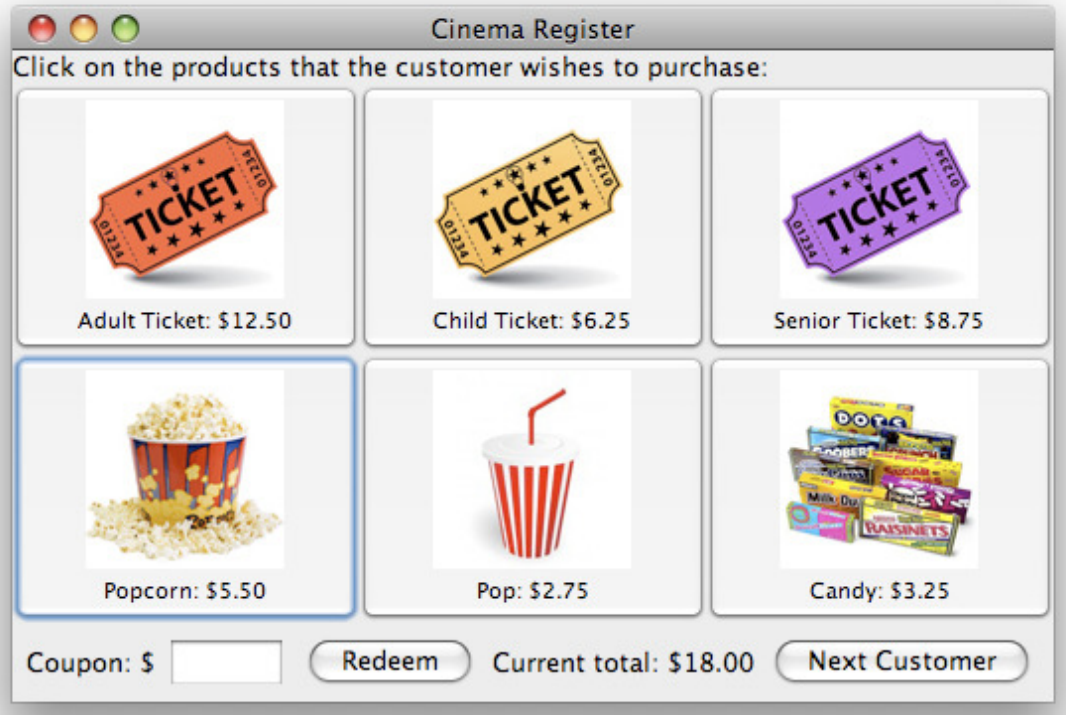
   Please follow these sample screen shots:



(GUI appearance when first loaded - no user interction yet).



(User presses the "Adult Ticket" button).



(User then presses the "Popcorn" button).

(User enters a coupon amount of $0.75 and presses the "Redeem" button. Note that the coupon amount is deducted from the total and the coupon amount field is cleared).

(User presses the "Next Customer" button).

Important: The components in your GUI must have the same arrangement as what is shown in the above screenshots. To accomplish this, you'll need to place your components on JPanels and choose appropriate layout managers. The following basic layout managers should suffice: BorderLayout, FlowLayout and GridLayout.

Use the prices and product names that are displayed in the screenshots above, and use the images that are contained in the .zip file from the start of this question.

Programming tip that you might find useful: When we talked about event handling in class, we mentioned the `getSource()` method. As you can see from the API documentation, this method returns a reference to the object on which the Event initially occurred. However, this is returned as an Object. If you know what type of object this is, you can cast it to that type; this will enable you to call (more specific) methods on that object. For example, if you knew that the source of the event was a JCheckBox, then you could do the following:

```
JCheckBox src = (JCheckBox)(e.getSource());
//You could then call a method from the JCheckBox class on src.
```

Once you are sure that your program is working correctly, print 5 screenshots: one screenshot before any buttons have been pressed, one after clicking on the "Adult Ticket" button, one after also clicking on "Child Ticket" and "Candy" each once and "Pop" twice, one after also redeeming a $2 coupon, and one after clicking on "Next Customer".

What to hand in to the assignment bin:

Submit paper copies of all source code (.java files) for question 1 and question 2 parts a and c, as well as the requested output (screen shots).

Assemble your printed pages in the same order as the questions. Attach the required cover page and **staple** together at the top left corner. Submit your assignment to the **correct** bin.

Important Note:

For this assignment, we also want you to submit your solutions electronically (via Desire2Learn). This will give the markers the opportunity to compile and run your programs (if they wish). To submit your assignment files electronically, please follow these steps:

- Create a .zip archive that contains all your source code (.java) files for question 1 and for question 2 parts a and c. (If you do not know how to create a zip file, please ask for help.)
- Name your .zip file in the following way: *YourName-as5.zip* For example: *JohnSmith-as5.zip*
- Log in to Desire2Learn and submit your.zip file to the dropbox that your instructor has set up for assignment #5.

End of Assignment 5

Maintained by Natalie Webber