

Homework 1: Movie Ticket Purchase

Date Due: August 31st, 2016

Write a program for Regal Entertainment Group that calculates the amount (in \$) the customers owe for purchase of movie tickets. First, your program should welcome the customer with some fancy message. Second, the program will ask the user to choose between two languages of your choice, validate for incorrect inputs (use while loop, allow for up to 5 incorrect answers, if the user enter incorrect answers 6 times, default to English). Display everything after this point in the chosen language. Then, the program must present the user with a menu with two options as follows (adjust to use appropriate language):

Please Enter One of the Following:

1. Purchase Tickets
2. Quit

Your Choice:

Validate for incorrect inputs (use while loops). Allow up to 5 incorrect answers. If the user enters incorrect answers 6 times quit the program. If the user enters option 1, prompt her for number of tickets she would like to purchase and if she is a member of the Regal Crown Club. And finally, display for user the amount she owes, including some fancy thank you message, keeping the following in mind: If user purchases 4 tickets or less, the cost of one ticket is \$10 each. For all purchases above 4 tickets (5 and above), the cost of one ticket is \$8 each. If the customer is a member of the Regal Crown Club, she receives an extra 5% discount and free small popcorn (make sure to display a message informing the customer of this). Enjoy!

Sample outputs (please don't make yours exactly like this one, use some creativity):

Welcome To Regal Entertainment Group.

Please Choose Your Language:

1. English

2. Spanish

Your choice: 1 [Enter]

Please Enter One of the Following:

1. Purchase Tickets

2. Quit

Your Choice: 1 [Enter]

How Many Tickets Would You Like to Purchase? 10 [Enter]

Are You a Member of the Regal Crown Club (y/n)? y [Enter]

Thank You for Your Purchase: Your Total is \$76.00. You Saved \$4.00 Today!!!

*** Free Small Popcorn ***

Redeem Entire Voucher at Concession Stand.

Submission guidelines: Please compress your Xcode project folder and name it **Ex01.-LastNameFirstNameInitial.zip** – (i.e. **Ex.01-ZejnilovicA.zip**) and submit it for grading in moodle.

Grading Rubric

	Unsatisfactory 0%	Satisfactory 100%
Readability (10% Weighting)	0 percent Program is NOT written neatly. At least one or more of the following is not true: * New line statements are indented 3-4 spaces. * Variables are declared correctly (start with lower case, use camelback notation) * Named constants are declared correctly (capitalized) * Comments are neatly indented	10 percent Program is written neatly: * New line statements are indented 3-4 spaces. * Variables are declared correctly (start with lower case, use camelback notation) * Named constants are declared correctly (capitalized) * Comments are neatly indented
Source Code (30% Weighting)	* Named constants used for values that do not change in the lifetime of the program * Variables used for everything that changes in the lifetime of the program. * Source code compiles and executes correctly. * Language menu is presented correctly * Payment menu options for each language implemented correctly * 10% discount calculation implemented correctly * 20% discount calculation implemented correctly * Logic to determine whether the user has enough funds implemented correctly.	* Named constants used for values that do not change in the lifetime of the program * Variables used for everything that changes in the lifetime of the program. * Source code compiles and executes correctly. * Language menu is presented correctly * Payment menu options for each language implemented correctly * 10% discount calculation implemented correctly * 20% discount calculation implemented correctly * Logic to determine whether the user has enough funds implemented correctly.
Logic (30% Weighting)	* The welcome message is implemented. * The logic to ask the user to choose between two languages along with validation of incorrect inputs (using a while loop, allowing for up to 5 incorrect answers) is implemented correctly. * If the user enters incorrect answers 6 times, the program defaults to English. Everything after this point is in the chosen language. * Appropriate discount is applied for purchase of more than 4 tickets	* The welcome message is implemented. * The logic to ask the user to choose between two languages along with validation of incorrect inputs (using a while loop, allowing for up to 5 incorrect answers) is implemented correctly. * If the user enters incorrect answers 6 times, the program defaults to English. Everything after this point is in the chosen language. * Appropriate discount is applied for purchase of more than 4 tickets
Submission (10% Weighting)	* The assignment was delivered as a compressed archive (.ZIP) and was attached to the DropBox. The assignment was named correctly per assignment requirements (Ex.01_LastNameFirstNameInitial.zip - i.e. Ex.01-ZejnilovicA.zip)	* The assignment was delivered as a compressed archive (.ZIP) and was attached to the DropBox. The assignment was named correctly per assignment requirements (Ex.01-LastNameFirstNameInitial.zip - i.e. Ex.01-ZejnilovicA.zip)
Delivery (20% Weighting)	The program was delivered after the due date (after the beginning of the class)	The program was delivered on due date by the beginning of the class.