At Chip’s Fast Food Emporium there is a very simple menu that offers a variety of food items ranging from healthy to unhealthy food choices. A calorie calculator is available for customer’s to help them make healthy choices. Each meal item is selected by entering the alpha character that corresponds with the menu option.

|  |  |
| --- | --- |
| **Burgers** | **Side Orders** |
| 1. Cheeseburger 461 Calories 2. Fish Burger 431 Calories 3. Veggie Burger 420 Calories 4. No burger 0 Calories | 1. Fries 100 Calories 2. Baked Potato 57 Calories 3. Chef Salad 70 Calories 4. No side order 0 Calories |
| **Drinks** | **Desserts** |
| 1. Soft Drink 130 Calories 2. Orange Juice 160 Calories 3. Milk 118 Calories 4. No drink 0 Calories | 1. Apple Pie 167 Calories 2. Sundae 266 Calories 3. Fruit Cup 75 Calories 4. No dessert 0 Calories |

Write a program that displays the menu options, prompts the user for their meal choices, computes the total calories for all the meal items selected, and displays a message about their food choices.

**Specifications**

The program will display the available menu choices with caloric values in tabular form. The user will be prompted to make a burger, a side order, a drink, and a dessert choice. Assume that the user will only enter valid menu choices (i.e. A, B, C, or D). However, the program must accept both uppercase and lowercase input for menu choices **(HINT: Use the equalsIgnoreCase method for String objects)**.

The program will also calculate and display the total calorie count for the user’s menu choices. If the calorie count is 900 or more, display the message “Your food choices are not healthy.” If the calorie count is 600 or more, display the message “Your food choices are moderately healthy.” Otherwise, display “Your food choices are healthy.”

**Sample Screen Layout**

Your screen must look like this for a user that selects a Fish Burger with Fries, Milk, and no dessert.

**Calorie Calculator for Chip’s Fast Food Emporium**

|  |  |
| --- | --- |
| **BURGERS** | **SIDE ORDERS** |
| 1. **Cheeseburger 461 Calories** 2. **Fish Burger 431 Calories** 3. **Veggie Burger 420 Calories** 4. **No burger 0 Calories** | 1. **Fries 100 Calories** 2. **Baked Potato 57 Calories** 3. **Chef Salad 70 Calories** 4. **No side order 0 Calories** |
| **DRINKS** | **DESSERTS** |
| 1. **Soft Drink 130 Calories** 2. **Orange Juice 160 Calories** 3. **Milk 118 Calories** 4. **No drink 0 Calories** | 1. **Apple Pie 167 Calories** 2. **Sundae 266 Calories** 3. **Fruit Cup 75 Calories** 4. **No dessert 0 Calories** |

**Please enter a burger choice: B**

**Please enter a side order choice: A**

**Please enter a drink choice: C**

**Please enter a dessert choice: D**

**Your total calorie count is 649.**

**Your food choices are moderately healthy.**

**Program Checklist**

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Marks** |
| **Knowledge** | **Programming Concepts:**   * Appropriate declaration of variables with meaningful names and  suitable data types * Appropriate use of object classes for input and output * Appropriate use of arithmetic operators to perform program calculations * Appropriate use of conditional statements and relational operators to evaluate program conditions |  |
| **Thinking** | **Algorithms:**   * IPO Chart, pseudo code, or flowchart provides detailed step-by-step instructions to properly implement the program specifications.   **~ OR ~**   * Detailed step-by-step instructions provided in the IPO Chart, pseudo code, or flowchart are accurately interpreted and implemented. |  |
| **Communication** | **Program Header:**   * contains programmer’s name, course code, date program written, and a comprehensive description of the purpose of the program   **Internal Documentation:**   * comments are used appropriately within the program and provide a meaningful summary of major processes   **Formatting:**   * program source code is properly indented where required and contains appropriate white space for readability * User interface is courteous, esthetically pleasing, and free of spelling and grammar errors |  |
| **Application** | **Implementation:**   * Tabular alignment of Chip’s Fast Food Emporium menu is formatted exactly as displayed in the sample layout. * Prompts for menu selections and user choices must be displayed on a single output line and formatted exactly as displayed in the sample layout. * Calculations to accumulate total calories for menu choices are accurate. * Evaluation of Boolean expressions yield the correct healthy food choice message. * Program source code is efficient and executes as required with no syntax or logic errors |  |