

## Homework 2 咖啡廳點餐系統

Deadline: 2019-11-1 23:59:59

### 前言

小明是一家咖啡廳的老闆，由於店裡人手不足，因此想利用點餐系統，來處理消費者的點餐以及結帳，這樣就可以將原先的外場人員，調度到內場廚房幫忙。

本作業希望同學以 Java 程式語言幫忙小明建立一個點餐系統，讓使用者能夠輸入新增餐點、查看以及修改已點選餐點等指令，同時計算出價格。此外，由於健康飲食是消費者注重的議題，因此系統會額外顯示熱量，讓消費者方便控制飲食。

### 作業需求

#### ■ Functional Requirements

- 使用者能點選菜單內的餐點。餐點、價格、熱量如下表所示：

品項		價錢(元)	熱量(大卡)
Americano	Medium	55	19.8
	Large	65	26.4
Cappuccino	Medium	70	217.6
	Large	85	290.6
Mocha	Medium	70	390.9
	Large	85	483.5
Latte	Medium	75	243.8
	Large	90	315.3
Macchiato	Medium	80	252.1
	Large	90	340.6

- 使用者可以調整咖啡的甜度，其指令如下：

- 無糖 no
- 微糖 low
- 半糖 half
- 少糖 less
- 全糖 normal

- 使用者可以調整咖啡的冰塊，其指令如下：

- 去冰 no
- 微冰 low
- 半冰 half
- 少冰 less
- 正常冰 normal

[備註]：為了降低作業的複雜度，因此暫不考慮熱咖啡

- 咖啡的甜度會增加額外的熱量，甜度和熱量的對照表如下：

甜度	熱量(大卡)
無糖 no	0
微糖 low	60
半糖 half	100
少糖 less	140
全糖 normal	180

- 結帳時，總價錢需計算同品項不限大小第二件半價，以**折價金額最少**的方式計算。遇見小數點時，對單一商品進行四捨五入，計算完再進行加總。

舉例：

1. 當使用者點選兩杯大杯 Cappuccino，一杯中杯 Cappuccino，則中杯的 Americano 以半價計算，總金額為  $85*2+70/2=170+35=205$  元
2. 當使用者點選三杯大杯 Americano 和兩杯中杯 Americano 時，計算方式為 (大杯 + 中杯) + (大杯 + 中杯) + 中杯，故兩杯中杯 Cappuccino 以半價計算，總金額為  $65*2+55+55/2+55/2=130+55+28+28=241$  元

## 基本指令集

指令格式	指令功能描述	限制
<b>add 飲品 大小 甜度 冰塊 份數</b>  範例： <b>add americano large half less 2</b>	新增咖啡至餐點明細中。 以範例來說，就是新增 2 杯大杯卡布奇諾半糖少冰至餐點明細中。	1. 份數的格式是阿拉伯數字 2. 份數需要大於 0 3. 每執行一次指令，就會在餐點明細新增一個編號，依照點餐順序，由 1 開始遞增
<b>check</b>	顯示目前所有的餐點明細，一則餐點明細的內容包含了編號、飲品、大小、甜度、冰塊、份數、價格以及熱量	1. 價格為一個編號的總價格，是飲品單價*份數 2. 熱量為單一飲品的熱量(飲料+甜度)，不乘以份數
<b>edit 編號 項目 內容</b>  範例： <b>edit 1 size medium</b>	可編輯某一則餐點明細中的大小(size)、甜度(sugar)、冰塊(ice)、份數(quantity)。 以範例來說，就是將第一則餐點明細中的大小改為 medium。	1. 份數的格式是阿拉伯數字 2. 份數需要大於 0
<b>menu</b>	顯示可選購的飲品，包含價格及熱量。	
<b>help</b>	列出所有指令集，並說明各個指令的功能	
<b>bill</b>	進行結帳動作，列出所有餐點明細和總價，再顯示感謝	

	用語，並結束系統。	
--	-----------	--

[備註]：

所有指令均是小寫或阿拉伯數字，指令和指令中間以半形符號的空白鍵區隔。

#### 小建議：

1. 進行本作業時，建議同學先思考要如何運用 Array 儲存資料，如何從 Array 中取用資料；哪些資料要放在一起，哪些要分開放，更動資料時哪些資料要一起更動。
2. 指令的輸入可以運用 String 的 Method: split() 實作，請參考 Additional Doc 的教學文件，並在 Related File 下載範例檔測試。

#### ■ Non-functional Requirement

- 提供完善、人性化、易閱讀的使用者介面
- 當使用者輸入指令後，需有相對應的回應，提供良好的使用者體驗
- 由於作業評分的需要，**不可**擅自增加額外的停頓點或延遲效果
- 當使用者輸入錯誤指令(包含拼字、格式、數字範圍錯誤等)，跳出錯誤提示並要求使用者重新輸入

#### ■ Execution Flow (Use Case)

1. 執程式後，系統顯示歡迎訊息並且顯示可選購的飲品，包含價格及熱量
2. 系統顯示所有可以使用的指令，並說明各個指令的功能，基本指令請參考「基本指令集」
3. 要求使用者輸入指令，系統依照使用者輸入的指令，執行對應的功能 (請參照基本指令集的指令功能描述)，完成後會要求使用者繼續輸入下一條指令 (除非上一條指令為 bill 指令)。
4. 若使用者輸入 bill 指令要進行結帳，系統會列出所有已點選的餐點明細和最終價錢，並且顯示感謝用語，結束本次操作。

### 額外加分項目

- **刪除功能**：指令格式「delete 編號」，系統會找到餐點明細中對應的編號，刪除該次點餐紀錄，此項目會在餐點明細中消失，在該編號後的餐點編號會自動向前補上，執行完後繼續要求使用者給予下一個指令。
- 在不破壞執行流程的限制下，可為此系統加入新功能或新特色以獲得額外加分，**但需在 readme.txt 中明確說明**，包含如何觸發或使用等等。
- User Interface 排列整齊且精美者，將能獲得額外加分。

### 作業繳交方式

請參考計概網站上的「作業繳交說明與規範」將作業相關檔案上傳到資料夾「hw2」當中。作業相關檔案分為以下四部分，括號中的數字為該部分的配分比：

1. **程式原始碼檔案 (70%)**：以 JAVA 根據作業要求撰寫而成的檔案，主程式檔名為「**hw2.java**」，主程式需撰寫註解幫助別人看懂你的程式碼，若同學的程式執行時需要用到其他檔案，如「ConsoleIn.java」，也需作為程式檔案的一部分一起繳交上來。

2. 文件檔案(27%)：檔名為「document.doc」、「document.docx」或「document.pdf」，文件需包含以下內容：
  - (1) 需求描述：描述使用者在使用此程式時可能會有何需求？你設計程式時如何考慮這些需求？程式中有哪些地方特別吸引使用者？
  - (2) 程式流程：說明程式進行流程，建議使用利用流程圖或是其他有助說明的圖示來幫助說明程式整體上如何運作
  - (3) 功能/邏輯說明：說明自己的程式裡，自己針對什麼樣的「功能」特地設計了什麼樣特別的「邏輯」來達成？建議用畫圖的方式來說明
  - (4) 使用說明：以教導使用者的角度，說明程式要如何執行使用或是需特別注意的點...等等。（不可只寫「看了就會使用」之類的話。）
  - (5) 其他：任何有助於別人了解或使用你的程式的說明。切勿流於閒聊
3. **readme.txt** (3%)：內容請按照「作業繳交說明與規範」當中的要求撰寫
4. **TimeLog.doc**：記錄此次作業撰寫時間。（「確實」填寫該表的同學可獲加 5 分）

## 特別注意事項

1. 繳交作業請使用 **SVN** 上傳，若有相關問題請參考計概網站上的 [計概帳號註冊/作業上傳教學](#)，上傳後可以使用論壇上「**check homework submission**」的功能進行確認
2. 請務必確保檔案擺放及命名符合規定，若作業檔案放錯資料夾、檔名大小寫或拼寫有誤，將可能造成作業無法正確批改。因此，**出現此情形將酌情扣分**
3. **作業請勿抄襲，所有作業皆會經過程式比對判斷是否抄襲，若發現一律以零分計算**