

# OOP 2023[F] Midterm

這份 Midterm 主要由國立臺北科技大學 109 資工系黃漢軒所命題。

## 題目敘述

Uriah 很不會喝酒，在第一次跑酒吧，跑去仁愛圓環附近的酒吧喝酒時，他認為啤酒喝起來很像汽油，然後妹酒喝起來很像魔術靈。

也還好他講話沒有那麼大聲，否則酒保可能會把 Uriah 的頭按進馬桶裡，然後讓 Uriah 發出咕嚕咕嚕的慘叫聲。

但 Uriah 注意到了這間酒吧的菜單很特別，他有啤酒、調酒與 Shot，且他們不是固定尺寸的杯子，例如啤酒可能是很大一杯，而 Shot 可能是很小一杯。

這讓 Uriah 開始想起了 OOP 課堂所教的一些觀念，他認為這麼多種類的啤酒、調酒與 Shot 應該能夠有效地被規劃成多個類別。

在這個 Task 中，給你 12 個已經部分實作好的類別，你需要額外實作各三個酒類的類別，並且你必須得要求自己制定滿足要求的繼承實作策略。

## 酒類表格

酒種	類別名稱	酒類	酒精濃度 ( 0% - 100% )	特別的配料	價格
馬丁尼	Martini	Cocktail	30%		500\$
瑪格麗塔	Margarita	Cocktail	30%	Lemon	400\$
梅酒	Mojito	Cocktail	30%		400\$
野火	Wildfire	Cocktail	30%	CinnamonWhiskey, AppleJuice	300\$
白蘭地杏仁	AmarettoSour	Cocktail	30%	OrangePiece	500\$
淡啤	Lager	Beer	5%		300\$
精釀啤酒	CraftBeer	Beer	5%		400\$
司陶特啤酒	Stout	Beer	5%		500\$

酒種	類別名稱	酒類	酒精濃度 ( 0% - 100% )	特別的配料	價格
啤酒花 IPA	IPA	Beer	5%		600\$
德國小麥啤酒	Hefeweizen	Beer	5%		400\$
火酒	Fireball	Shot	20%		800\$
牛仔特調	CowboyShot	Shot	30%		600\$
金酒	Goldschlager	Shot	20%		200\$
龍舌蘭酒	Tequila	Shot	40%		800\$
蘭姆酒	Rum	Shot	60%		800\$

對於以上的酒種，由於某些餐廳可能會期望調酒、Shot 可能會根據這些酒種進不同的酒精濃度，因此我們也期望這些酒精濃度能夠被調整。

並且我們準備了一個酒類品質指標的計算公式，你應該也要處理這件事情。

請查看任務描述與任務階段來確認應該要執行哪些事情。

## 任務描述

對於每個酒類，必須要是使用 Default Constructor 來建構，並且建構後能夠取得上表所描述的數值，例如：

```
Mojito mojito; // 使用 Default Constructor 來建構
mojito.GetPrice(); // 價格為 $400
```

詳情可見 `ut_CheckPoint1.cpp`、`ut_CheckPoint2.cpp` 與 `ut_CheckPoint3.cpp` 的測試案例。

當你能夠瞭解這件事情後，你可以開始依照以下的步驟進行酒類的設計。

我們強烈建議你按照這三個步驟將期中考完成，但如果你認為你應該可以省略某些步驟來讓自己更快完成期中考，你可以選擇跳過某些步驟。

## 第一步驟：實作三種酒類

在這個題目中，我們已經完成了 15 種酒類中的 12 種酒類。

你需要額外撰寫缺少的三個酒類，分別是梅酒 ( Mojito )、斯陶特啤酒 ( Stout )、以及蘭姆酒 ( Rum )。

在這個任務中，你需要撰寫這三個類別，並且使這三個類別具有取得酒類名稱的 Getter `GetType()` 能夠回傳該三種酒類的酒種是什麼。

## 第二步驟：實作酒類基礎功能

我們期望所有的酒類都應該要滿足以下的功能：

- 對於每個酒種，能夠回傳對應的酒類名稱，例如：蘭姆酒應有一個 Getter `std::string GetType()` 來回傳酒類名稱。
- 對於每個酒種，能夠回傳對應的酒精濃度：例如：蘭姆酒應有一個 Getter `int GetAlcoholContent()` 來回傳酒精濃度。
- 對於每個酒種，能夠回傳對應的價格，例如：蘭姆酒應有一個 Getter `int GetPrice()` 來回傳該酒種的價格。
- 對於調酒，他們必須要有自己額外的配料，例如：蘭姆酒應有一個 Getter `std::vector<std::string> GetTopping()` 來取得配料 `vector`。
- 對於啤酒，他們的酒精濃度應該要固定成 5%。
- 對於調酒，他們的酒精濃度應該要固定成 30%。

你可以自由定義成員與方法，僅需確保能夠滿足以上描述的功能。

## 第三步驟：實作酒類進階功能

我們期望所有的酒類都應該要滿足以下的功能：

- 調酒與 Shot 應可以調整自己的酒精濃度，你應該實作一個 Setter `void SetAlcoholContent(int alcoholContent)` 來設定或取得調酒的酒精濃度。
  - 酒精濃度應該要被限制，如果設定的濃度大於 100，你應該要拋出 `std::invalid_argument` 例外。
- 你需要製作一個酒類的品質指標  $Q_S$  計算 Getter `int GetQuality()`，如下：
  - 基礎分數  $B_S$ ：調酒 10、啤酒 8、Shot 7。
  - 酒精分數  $A_S$ ：等於酒精濃度 (%)。
  - 配料分數  $T_S$ ：配料是否存在，如果配料存在，則分數為 10，否則為 5。
  - 價格分數  $P_S$ ： $price/100 \times 10$ 。
  - 酒類品質指標  $Q_S$ ： $30 \times B_S + 25 \times A_S + 20 \times T_S + 25 \times P_S$

你可以自由定義成員與方法，僅需確保能夠滿足以上描述的功能。

## 任務備註

- 你應該要制定一個良好的繼承策略來能夠處理這大量的實作。
- 你不用額外撰寫測試，不用撰寫測試來達到 95% 的覆蓋率。
- 我們有提供給你們測試，你們可以自行開啟它來進行測試。

## 任務配分

本試卷將依照通過測試的案例比例進行配分。

- [ 30% ] 實作三種酒類所得到的基礎分數
- [ 30% ] 實作酒類基礎功能分數
- [ 40% ] 實作酒類進階功能分數

## 注意事項

- 若你想要有空指標，請將指標賦值（assign）`nullptr`，這個特殊的指標可用於指向不存在的東西。
- 對於 Google Test 的測試函數：
  - 使用 `ASSERT_EQ` 來測試不會有精度誤差的值（例如：整數、字串等）
  - 使用 `ASSERT_NEAR` 來測試具有精度的值
  - 使用 `ASSERT_THROW` 來測試該函數會不會拋出例外
  - 使用 `ASSERT_FALSE` 來測試該值是否為 `false`
- 以下情況可能導致你的期中考試成績出現問題：
  - 你的專案不應該出現 Memory Leak，否則將會扣除考試總分 10 分。
  - 你不應該上傳 `/bin` 資料夾至專案庫，編譯結果不應該上傳至專案庫上。
    - 若在助教確認考試評分時 `/bin` 資料夾存在在專案庫（GitLab）中，扣除考試總分 10 分。
- 你可以使用 `debugger` 工具來進行除錯，有利於尋找漏洞與理解漏洞為何會發生。

## Meme

HW3 忘記放梗圖了，補一張：)



