

CMSC 202 Spring 2023

Project 2 – Potion Craft

Assignment: Project 2 – UMBC Potion Craft

Due Date: Tuesday, March 14th at 8:59pm on GL

Value: 80 points

1. Overview

In this project, you will:

- Practice basic C++ syntax including branching structures,
- Write classes and instantiate those classes using a constructor,
- Use arrays to hold objects,
- Use simple file input,
- Practice breaking projects into multiple files,
- Use a makefile to compile a project.

2. Background

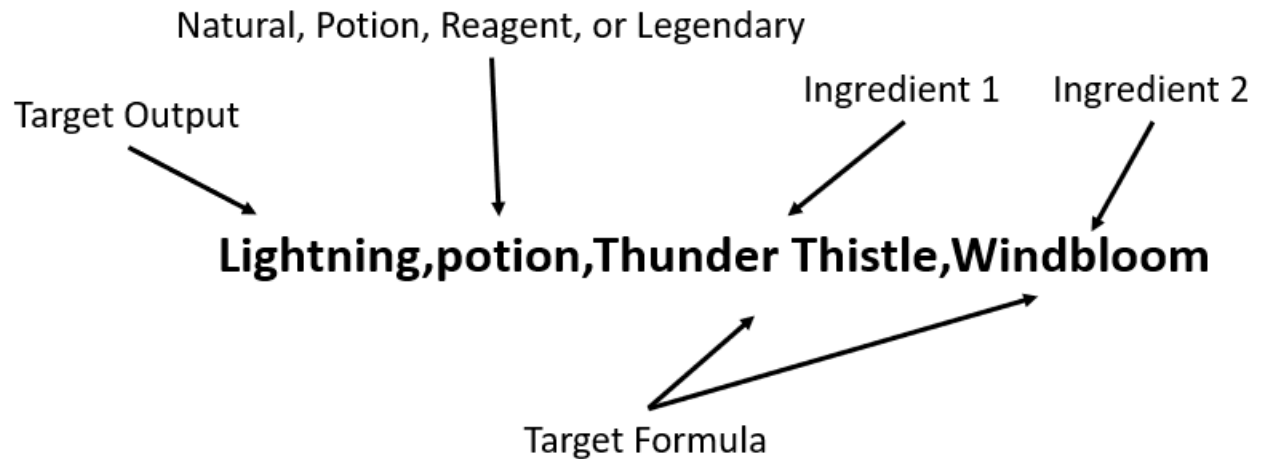
Potion Craft is an alchemist simulator where you mix ingredients to brew potions. There are several ways you can gather the required ingredients for a potion. You might go out into nature and search for some plants. After searching for some natural ingredients, maybe you will grind ingredients and carefully mix them in your cauldron. Heat the coals. Boil and stir. Add the base: water, oil, or... something else. Part of the fun for this type of game is that you will need to experiment to find new recipes.

You start the game after you have just opened your new potions shop. Unfortunately, you don't start with anything so you will have to go out and forage for ingredients. As you start to build a stockpile of ingredients, you can try and make more complex potions. Ultimately, you are trying to build the ultimate shop where someone could buy any natural ingredients, potions, reagents, or legendary products!

Item Data:

- Natural (Natural ingredients, these are just found by randomly searching)
- Potion (crafted from basic items - natural ingredients, can't be found)
- Reagent (crafted from a mix of natural ingredients and potions. Can't be found)
- Legendary (crafted from only reagents and potions, can't be found)

Here is an example from the input file. This says, “to make Lightning, which is a potion, I will need 1 Thunder Thistle and 1 Windbloom”.



How to rank up your shop:

When you start the game, you have just opened your potions shop and your shop will start at rank F. Your goal is to build up your available products until you have all products available and in stock (quantity > 0). First, you try and build up the number of natural ingredients you have. Then you can build some potions. As you continue to build your stockpile, your shop's rank will improve. Once you get 100% of the possible natural ingredients, potions, regents, and legendary items, you have won the game!

- F Rank (starting rank)
- F Rank -> D Rank (You need $\geq 30\%$ of all available items)
- D Rank -> C Rank (You need $\geq 50\%$ of all available items)
- C Rank -> B Rank (You need $\geq 70\%$ of all available items)
- B Rank -> A Rank (You need $\geq 90\%$ of all available items)
- A Rank -> S Rank (To win, you need 100% of all available items)

3. Assignment Description

Initially, you will need to read in a list of ingredients and recipes from a file and load them into an array of ingredients. The list of ingredients is static, and you can assume that the size can be stored in a constant (PROJ2_SIZE).

Your shop does not start with anything, but you can begin searching for natural ingredients immediately. Anytime a natural ingredient is found, it is added to the shop's collection of items. Once the player acquires at least 2 natural ingredients, they may begin crafting new potions, reagents, or legendary items by merging the ingredients. For example, Waterbloom and Terraria can be combined to make a Healing potion. All items that can be randomly acquired are considered natural ingredients. Other items such as potions, reagents, and legendary items are made of combinations of other ingredients.

4. Requirements:

This is a list of the requirements of this application. For this project, you will be provided with header files to start you in the right direction. For you to earn all the points, however, you will need to meet all the defined requirements.

- You must follow the coding standard as defined in the CMSC 202 coding standards (found on Blackboard under course ingredients). This includes comments as required.
- The project must be turned in on time by the deadline listed above.
- The project must be completed in C++. You may not use any libraries or data structures that we have not learned in class. Libraries we have learned include `<iostream>`, `<fstream>`, `<iomanip>`, `<vector>`, `<cmath>`, `<ctime>`, `<cstdlib>`, and `<string>`. You may not use vectors – everything must be implemented in arrays. You should only use `namespace std`.
- Using the provided files, `Ingredient.h`, `Shop.h`, `Game.h`, `makefile`, `proj2_data.txt` and `proj2.cpp`, create UMBC PotionCraft. You can copy the files from my directory in `/afs/umbc.edu/users/j/d/jdixon/pub/cs202/proj2`.
- To copy them, navigate to your project 2 folder and type:

```
cp /afs/umbc.edu/users/j/d/jdixon/pub/cs202/proj2/* .
```
- You **must** use the provided header files (`Ingredient.h`, `Shop.h` and `Game.h`). Do not add or change any constant, variables, or functions in these files. Do not add member variables to any class.
- All user input must be validated. For example, if a menu allows for 1, 2, or 3 to be entered and the user enters a 4, it will re-prompt the user. However, the user is expected to always enter the correct data type. i.e. If the user is asked to enter an integer, they will. If they are asked to enter a character, they will. **You do not need to worry about checking for correct data types.**
- There is a single input file for this project named, `"proj2_data.txt"`. The file name can be stored as a constant. The file is already provided in Prof. Dixon's course folder on GL.
- For this project, **pointers are not to be used**, however, pass-by-reference is used in a function.
- The player's name can have a space (use `getline`). Additionally, the names of the ingredients may have a space.
- There are four types of ingredients:
 1. Natural ingredients that can be randomly found by searching.
 2. Potions can be created by merging two natural ingredients.
 3. Reagents can be created by merging two potions, or a different reagent and a potion.
 4. Finally, legendary items can be made from merging a reagent and a potion.
- Have a main menu that asks if the user wants to:
 1. Display current ingredients and their current quantities in a numbered list.
 2. Search for Natural Ingredients
 - Randomly finds one of the natural ingredients from the list of natural ingredients. The ingredient must have the type of "natural" and should not have any required ingredients. Increases the quantity in the Shop's inventory.

3. Merge Ingredients

- Asks which ingredients you would like to combine. Checks a few things: 1. Validates that the numbers entered are within the range of available ingredients. 2. Checks to see if there is a recipe with those two ingredients. 3. Checks to see if there is enough quantity to merge those two ingredients. 4. Decrements the two ingredient quantities and increments the created product.
 - For example, if you were trying to merge a Waterbloom and Terraria, you would check to see if there is a recipe for merging those two ingredients. If there is, you would then check to see if the shop has at least one of each of those ingredients in stock. If it does, it decrements each of those ingredients and increments the healing potion.
- A recipe for any item can be made up of two ingredients in **any** order. For example, a poison potion could be made by merging Terraria and Firebell OR by merging Firebell and Terraria.
- `RequestIngredient()` inside of the Game class is simply the function that asks the user which ingredient they would like to merge. It does something specific if the user enters a -1 (which should display all ingredients) and it validates to make sure they enter a valid choice. The player needs to have at least 1 quantity of that ingredient to use it in a combination. For example, to make Poison, the player must have at least one Terraria and at least one Firebell.

4. Display Score

- The player wins the game once their shop reaches S rank, requiring at least one of each ingredient, potion, reagent, and legendary.

5. Exit

5. Recommendations

You must use the provided header files (`Ingredient.h`, `Shop.h`, and `Game.h`) additionally, we provided you with the `makefile` and the `proj2.cpp`.

Here are some general implementation recommendations (do not follow these verbatim – these are GENERAL suggestions):

- Read each of the header files in detail. Read through the project document in detail. Use paper to take notes. Each header file has tons of comments.
- Design the solution (part is already designed, so make sure you understand how it works) on PAPER.
- Read through the provided `Ingredient.h`. Think about how you can use the starting file `proj2_data.txt` to populate an array of ingredients.
- Start with the `Shop.cpp` file and code everything. The Shop class only uses `cout` statements in the `ShopScore` function (no other function should have a `cout` statement in `Shop.cpp`). Don't forget the constructor! Test everything incrementally.
- Start `Game.cpp` – start building the constructor. Incrementally build it as you are testing a function.

- Once `Shop.cpp` is written, start on `Game.cpp` – start with loading in the ingredients from the `proj2_data.txt` file. This will be tricky because you have ingredient names with spaces and a comma delimited data file. Check out the **delimiter** feature in `getline` here: <https://www.geeksforgeeks.org/getline-string-c/>
- `CombineIngredients()` is the most difficult function to write so expect to spend some time finishing it. Don't forget that the ingredients can be in either order (A+B) or (B+A).
- After the files are successfully loaded (and displayed) work on the main menu.

6. Sample Input and Output

For this project, the input file is very simple. The only input file is called `proj2_data.txt`. You can code the file name as a constant.

The columns for `proj2_data.txt` are as follows: ingredient name, ingredient type (natural, potion, reagent, or legendary), first ingredient required to make, and second ingredient required to make.

Natural ingredients have no first and second ingredients to make. They can only be found by searching.

Potions and reagents are an intermediate ingredient that are made by combining two natural or common ingredients.

Legendary ingredients are a final ingredient that are created by combining two intermediate ingredients (potions and reagents).

For example, Windbloom is Natural and can be found by searching (option 2 in the main menu). A Swift potion can be made by merging two windblooms (assuming they are in inventory). These combinations continue until you have at least one of each ingredient in inventory with a quantity greater than zero.

Here is part of the input file (there are 32 recipes in the file).

```
Firebell,natural,none,none
Waterbloom,natural,none,none
Terraria,natural,none,none
Windbloom,natural,none,none
Mad Mushroom,natural,none,none
Witch's Mushroom,natural,none,none
Dexterity,potion,Waterbloom,Windbloom
Light,potion,Windbloom,Firebell
Rage,potion,Bloodthorn,Featherbloom
Charm,potion,Mad Mushroom,Windbloom
Levitation,potion,Mad Mushroom,Witch's Mushroom
Invisibility,potion,Witch's Mushroom,Windbloom
Nigredo,reagent,Poison,Stone Skin
Albedo,reagent,Nigredo,Frost
```

```
Citrinitas, reagent, Albedo, Fire  
Rubedo, reagent, Citrinitas, Healing  
Philosopher's Stone, legendary, Rubedo, Invisibility
```

The file can be downloaded from Prof. Dixon's data folder by navigating to your project 2 folder and typing the following command:

```
cp /afs/umbc.edu/users/j/d/jdixon/pub/cs202/proj2/proj2_* .
```

After you copy the data file, you can type “`cat proj2_data.txt`” and it should show you the entire data file.

In the sample output below, user input is colored blue for clarity. After compiling and running proj2, the output would look like this:

```
[jdixon@linux2 proj2]$ make run  
./proj2  
**Potion Craft Art from GameTitle**  
32 ingredients loaded.  
What is the name of your Shop?  
JD  
What would you like to do in JD's shop?  
1. Display your Shop's Ingredients  
2. Search for Natural Ingredients  
3. Attempt to Merge Ingredients  
4. See Score  
5. Quit  
5  
Thanks for playing UMBC Potion Craft
```

If you were to display Shop's Ingredients, it would look like this:

```
What would you like to do in JD's shop?  
1. Display your Shop's Ingredients  
2. Search for Natural Ingredients  
3. Attempt to Merge Ingredients  
4. See Score  
5. Quit
```

```
1
1. Firebell 0
2. Waterbloom 0
3. Terraria 0
4. Windbloom 0
5. Mad Mushroom 0
6. Witch's Mushroom 0
7. Thunder Thistle 0

**8-26 removed for formatting**

27. Invisibility 0
28. Nigredo 0
29. Albedo 0
30. Citrinitas 0
31. Rubedo 0
32. Philosopher's Stone 0

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
```

If the user would choose 2, Search for Natural Ingredients, then the output would look like this:

```
What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit

2

Bloodthorn Found!
```

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit

2

Waterbloom Found!

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit

2

Windbloom Found!

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit

2

Bloodthorn Found!

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit

If the user would choose 3, Attempt To Merge Ingredients, then choose something that we have in inventory:

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients


```
4. See Score
5. Quit
3
Which ingredients would you like to merge?
To list known ingredients enter -1
2
Which ingredients would you like to merge?
To list known ingredients enter -1
4
Waterbloom combined with Windbloom to make Dexterity!
You have made Dexterity for your shop.
What would you like to do in JD's shop?
1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
```

Above we created a Dexterity potion from Waterbloom and Windbloom. The quantities for both Waterbloom and Windbloom would be reduced by one and the quantity for Dexterity would increase by one.

Below is an example where we try to make a Fire potion from two Firebells but we don't have the quantity to make that merge.

```
What would you like to do in JD's shop?
1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
3
Which ingredients would you like to merge?
To list known ingredients enter -1
1
Which ingredients would you like to merge?
To list known ingredients enter -1
1
You do not have enough Firebell or Firebell to attempt that merge
What would you like to do in JD's shop?
```

```
1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
```

Here is an example where the See Score is selected. As the number of items with quantity increases, shop's rank should go up. Once 100% of the available natural ingredients, potions, reagents, and legendary items have been created, the game ends.

```
What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit

4

***The Potion Shop***

Owned by: JD

There are 32 available products.

This shop has 0 products in stock.

Which is 0%

Shop Rank: F

*****

What would you like to do in JD's shop?

1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
```

Here are some example runs where additional input validation is being shown and where the name of the Shop name has a space:

```
What is the name of your Shop?

Michael Myers

What would you like to do in Michael Myers's shop?

1. Display your Shop's Ingredients
```

```
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
0
What would you like to do in Michael Myers's shop?
1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
6
What would you like to do in Michael Myers's shop?
1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
4. See Score
5. Quit
3
Which ingredients would you like to merge?
To list known ingredients enter -1
0
Which ingredients would you like to merge?
To list known ingredients enter -1
100
Which ingredients would you like to merge?
To list known ingredients enter -1
1
Which ingredients would you like to merge?
To list known ingredients enter -1
1
You do not have enough Firebell or Firebell to attempt that merge
What would you like to do in Michael Myers's shop?
1. Display your Shop's Ingredients
2. Search for Natural Ingredients
3. Attempt to Merge Ingredients
```

```
4. See Score
5. Quit
5
Thanks for playing UMBC Potion Craft**5-61 removed for space**
```

There is a longer run of the game available as `proj2_sample1.txt` with the other files.

7. Compiling and Running

We have provided you with a sample `makefile` which should help you compile all of the classes and the program itself.

Once you have compiled using the provided `makefile`, enter the command `make run` or `./proj2` to run your program. If your executable is not `proj2`, you will lose points. It should look like the sample output provided above.

8. Completing your Project

When you have completed your project, you can copy it into the submission folder. You can copy your files into the submission folder as many times as you like (before the due date). We will only grade what is in your submission folder.

For this project, you should submit the following files to the `proj2` subdirectory:

`Shop.cpp`, `Shop.h`

`Game.cpp`, `Game.h`

`Ingredient.h`, `proj2.cpp`

For this project, you should only modify the header files to add helper functions but do not add any additional variables (such as `Shop.h` or `Game.h`). Almost all edits should be in the classes `.cpp` files (`Shop.cpp` and `Game.cpp`).

You do not need to submit the `makefile`.

As you should have already set up your symbolic link for this class, you can just copy your files listed above to the submission folder.

You can also make a rule in your `makefile` to copy all of the files using `make submit` if you would like (although it is not provided).

A. `cd` to your project 2 folder. An example might be `cd ~/202/projects/proj2`

B. `cp Ingredient.h Shop.h Shop.cpp Game.h Game.cpp proj2.cpp ~/cs202proj/proj2`

You can check to make sure that your files were successfully copied over to the submission directory by entering the command:

```
ls ~/cs202proj/proj2
```

Make sure that the required files are submitted by the deadline. If the copy command provided does not work, it is your responsibility to figure out what is wrong and that all required files have been submitted.

You can check that your program compiles and runs in the `proj2` directory, but please clean up any `.o` and executable files. Again, do not develop your code in this directory and you should not have the only copy of your program here. Uploading of any `.gch` files will result in a severe penalty.

IMPORTANT: If you want to submit the project late (after the due date), you will need to copy your files to the appropriate late folder. If you can no longer copy the files into the `proj2` folder, it is because the due date has passed. You should be able to see your `proj2` files but you can no longer edit or copy the files in to your `proj2` folder. (They will be read only)

- If it is 0-24 hours late, copy your files to `~/cs202proj/proj2-late1`
- If it is 24-48 hours late, copy your files to `~/cs202proj/proj2-late2`
- If it is after 48 hours late, it is too late to be submitted.

9. List of Possible Ingredients

Below is a list of all possible ingredients that can be created using merge but this is all available in the `proj2_data.txt` file. You can write down combinations to help test your code more efficiently.

```
Firebell,natural,none,none
Waterbloom,natural,none,none
Terraria,natural,none,none
Windbloom,natural,none,none
Mad Mushroom,natural,none,none
Witch's Mushroom,natural,none,none
Thunder Thistle,natural,none,none
Lifeleaf,natural,none,none
Bloodthorn,natural,none,none
Featherbloom,natural,none,none
Healing,potion,Waterbloom,Terraria
Poison,potion,Terraria,Firebell
Frost,potion,Waterbloom,Waterbloom
Fire,potion,Firebell,Firebell
Explosion,potion,Mad Mushroom,Mad Mushroom
Swiftess,potion,Windbloom,Windbloom
Mana,potion,Witch's Mushroom,Witch's Mushroom
Lightning,potion,Thunder Thistle,Windbloom
Strength,potion,Terraria,Terraria
Stone Skin,potion,Firebell,Waterbloom
```

Sleep,potion,Lifeleaf,Waterbloom

Dexterity,potion,Waterbloom,Windbloom

Light,potion,Windbloom,Firebell

Rage,potion,Bloodthorn,Featherbloom

Charm,potion,Mad Mushroom,Windbloom

Levitation,potion,Mad Mushroom,Witch's Mushroom

Invisibility,potion,Witch's Mushroom,Windbloom

Nigredo,reagent,Poison,Stone Skin

Albedo,reagent,Nigredo,Frost

Citrinitas,reagent,Albedo,Fire

Rubedo,reagent,Citrinitas,Healing

Philosopher's Stone,legendary,Rubedo,Invisibility