

## Midterm Project: Lost and Found Inventory Tracker:

For my Midterm Project I wanted to develop a Lost and Found Inventory Tracker!

**Project:** Lost and Found Inventory Tracker

**Project Scope:** This program is a simple Lost & Found Inventory Tracker where users can log items they've found and search for them later. When you run the program, it first shows you the sizes of common data types in C++ and will then prompt you to enter five items—each with a name, ID number, and whether it's valuable or not. This will then prompt a full list of inventory in the system and then you can even search for an item using its unique ID number. If there is an item with that ID number in the system, then you will see the name of the item, its corresponding ID number, and also if it's listed as valuable or not. It will also log that information into a transactions.txt file as well. This was an idea based off of the requirements that were listed in the directions!

## Midterm Project Requirements & Implemented:

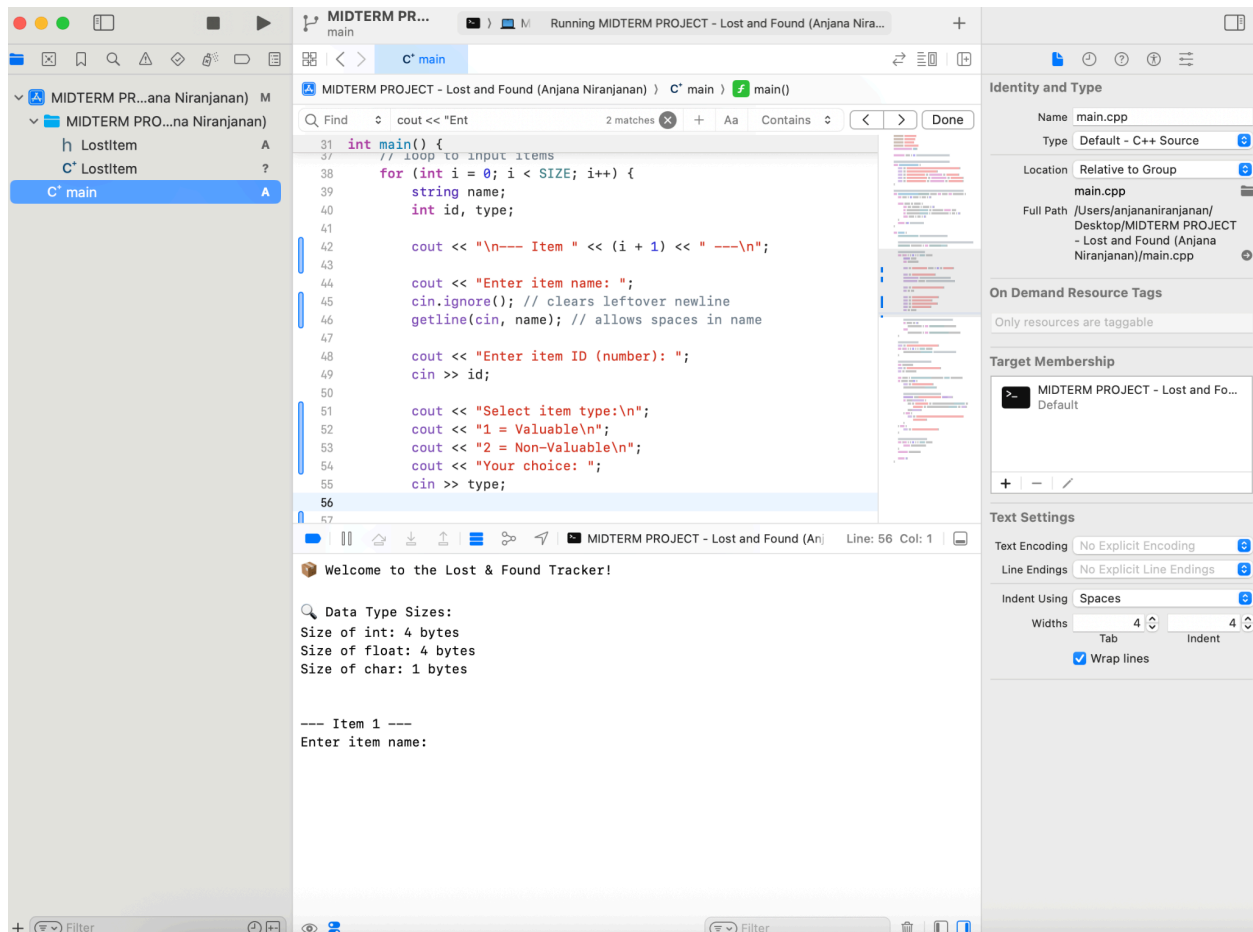
Below shows my thought process and how I satisfied each of the project requirements while also labeling which assignments I referred to in order to code my project based off of what we have done in this class so far:

- Data types and data sizes
  - I referenced Assignment 1. For the start of my program I included a function that prints the sizes of data types we covered such as int, char, and float.
- File types (e.g., .txt for data I/O)
  - I referenced the file handling example from Assignment 6 and 8, and used a .txt file named transactions.txt to log items that were successfully claimed
- Use of pointers
  - I used dynamic memory to keep track of lost items with pointers, similar to what we did in Assignment 5.
- Arrays
  - I made an array with 5 spots to store lost items, like we did in Assignment 9 when we worked with class objects in arrays.
- A binary search algorithm (on any other search)
  - Similar to what we did in Assignment 5, I used a binary search to look up items by their ID to check if something was in the tracker.
- Use of strings
  - I used string variables to hold and show the names of each item which we learned in Assignments 2 and 6.
- File I/O (read from or write to a file)
  - I used ofstream to save item info into a transactions.txt file when an item was found, similar to how we saved data in Assignment 6.

- At least two classes (with meaningful interaction)
  - I created a base class called `LostItem` and two derived classes `ValuableItem` and `NonValuableItem` based on the abstract class structure from Assignment 7, and they work together using polymorphism.

## Midterm Project Outputs & Screenshots:

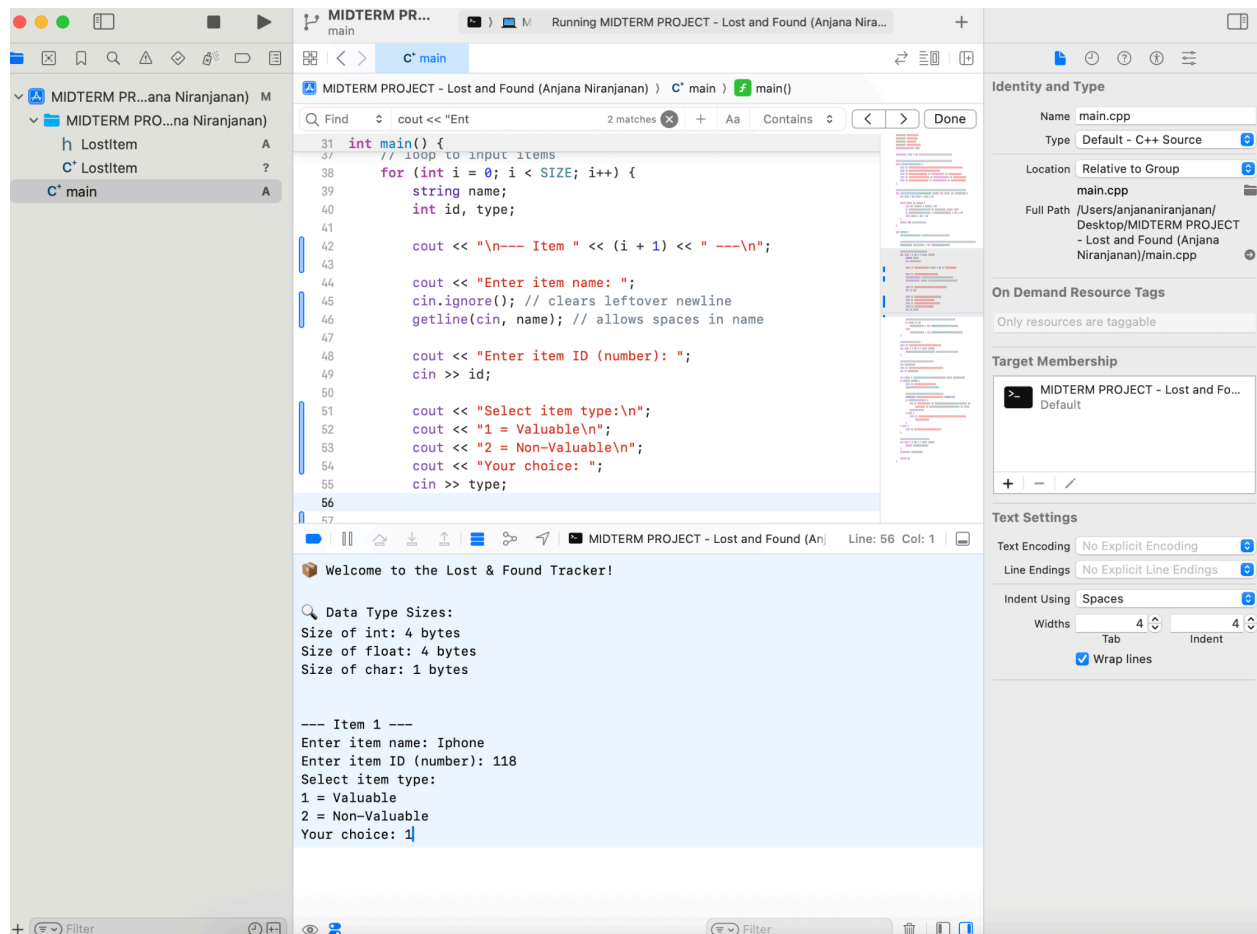
### Screenshot 1:



Screenshot 1 (above) displays:

- A welcome message
- Sizes of key data types
- Prompts the user to enter the first item's name
  - Note that when the first item name is entered, then the next respective question: "ID Number" will show up on the console and then the next: "Valuable or Non-Valuable".

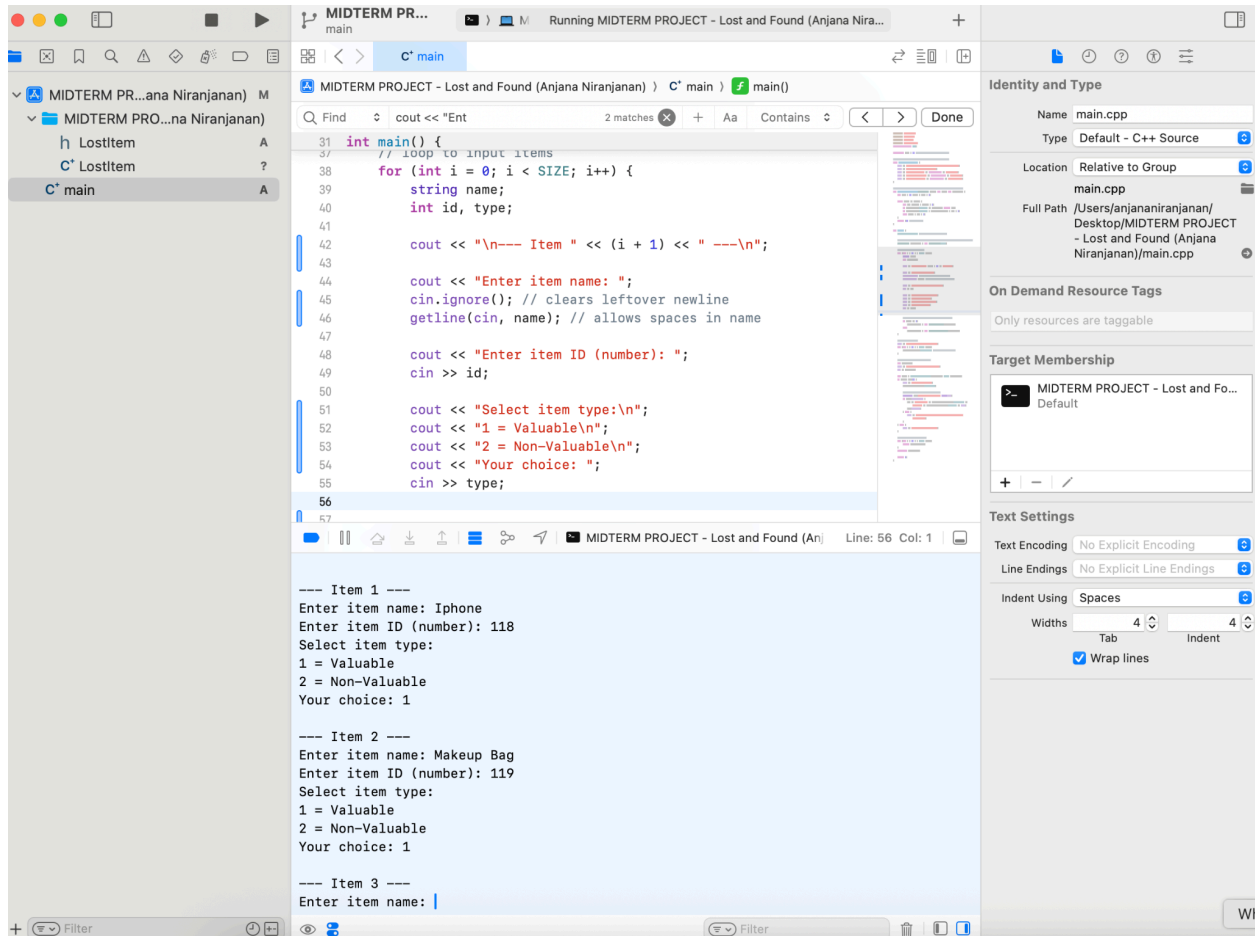
## Screenshot 2:



Screenshot 2 (above)

- The user enters the first Item's information which shows how the program collects and classifies each lost item
  - Name: "iPhone"
  - ID: 118
  - Type: 1 (Valuable)

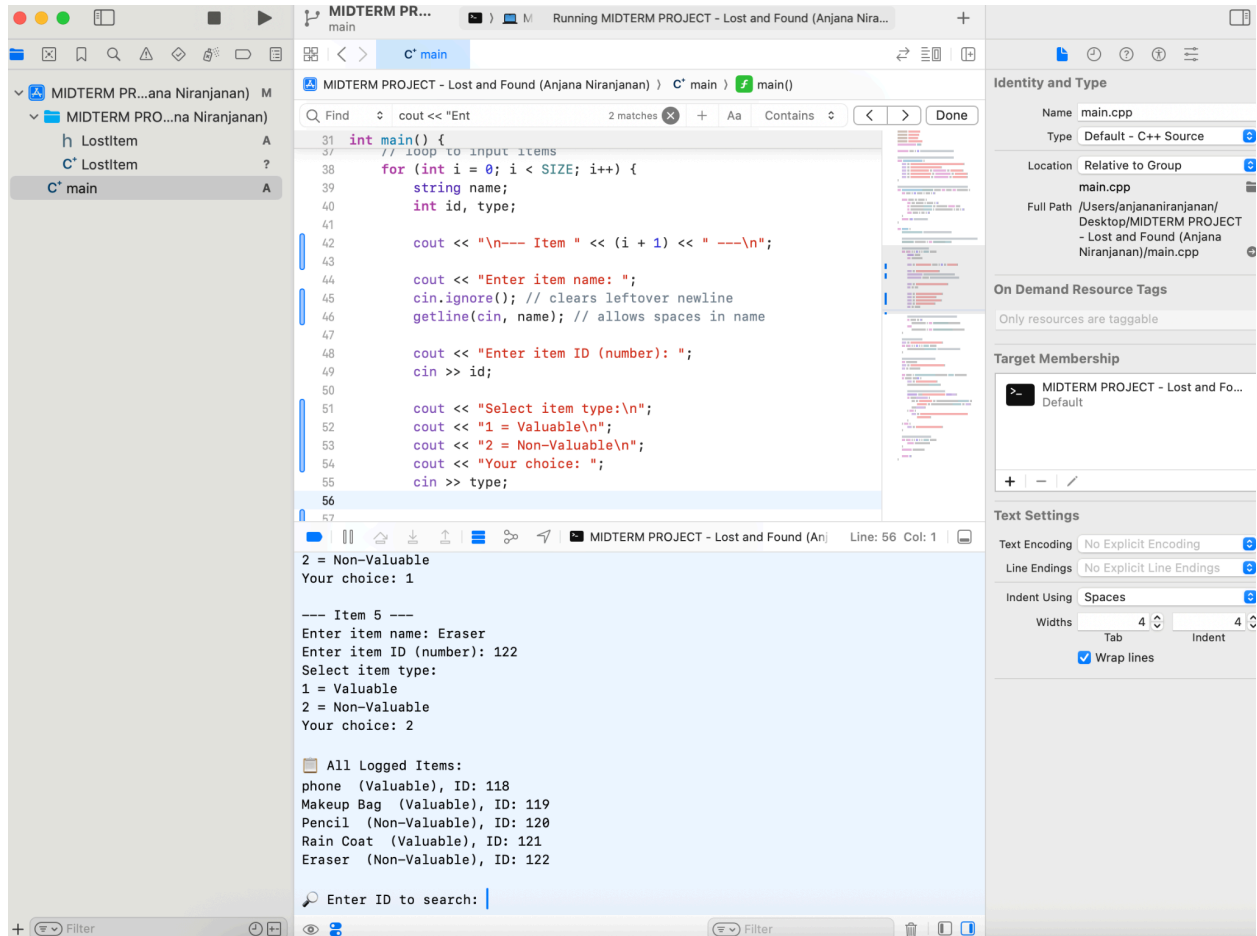
### Screenshot 3:



Screenshot 3 (above)

- Input continues for more items and shows the second item: "Makeup Bag" being inputted
- The prompt appears for the third item, and this process will repeat until the fifth item is entered.

## Screenshot 4:



Screenshot 4 (above)

- User finishes entering five total items (both valuable and non-valuable)
- The program will then display a full inventory of all logged items
- Prompts the user to **search by ID** for an item in the tracker

## Screenshot 5:

```
#include "LostItem.h"
using namespace std;

const int SIZE = 5;

void printDataSizes() {
    cout << "👋 Welcome to the Lost & Found Tracker!\n";
    cout << "\n🔍 Data Type Sizes:\n";
    cout << "Size of int: " << sizeof(int) << " bytes\n";
    cout << "Size of float: " << sizeof(float) << " bytes\n";
    cout << "Size of char: " << sizeof(char) << " bytes\n";
}

int binarySearchById(LostItem** arr, int size, int targetId) {
    int left = 0, right = size - 1;
    while (left <= right) {
        int mid = (left + right) / 2;
        if (arr[mid]->getId() == targetId)
            return mid;
        else if (arr[mid]->getId() < targetId)
            left = mid + 1;
        else
            right = mid - 1;
    }
    return -1;
}
```

Enter item ID (number): 122  
Select item type:  
1 = Valuable  
2 = Non-Valuable  
Your choice: 2

All Logged Items:  
phone (Valuable), ID: 118  
Makeup Bag (Valuable), ID: 119  
Pencil (Non-Valuable), ID: 120  
Rain Coat (Valuable), ID: 121  
Eraser (Non-Valuable), ID: 122

🔍 Enter ID to search: 121  
✅ Item found:  
Rain Coat (Valuable), ID: 121  
📄 Item written to transactions.txt  
Program ended with exit code: 0

Screenshot 5 (above)

- The user searches for an item with an ID of 121 so the user can see details about that respective item.
  - The item ("Rain Coat") is found and displayed
- A record is saved to transactions.txt
- The program is completed!!!

**Challenge #1 :** At first, the file wasn't saving anything even after a successful item search and then I realized it was because the code to write to the file came after the item was found, so I made sure that block only runs if the item exists.

**Challenge #2:** Another issue was that transactions.txt wasn't even showing up in the project folder. So, I ran the program once completely and realized the file only gets created after an item is actually searched and logged and when I did that the file appeared.