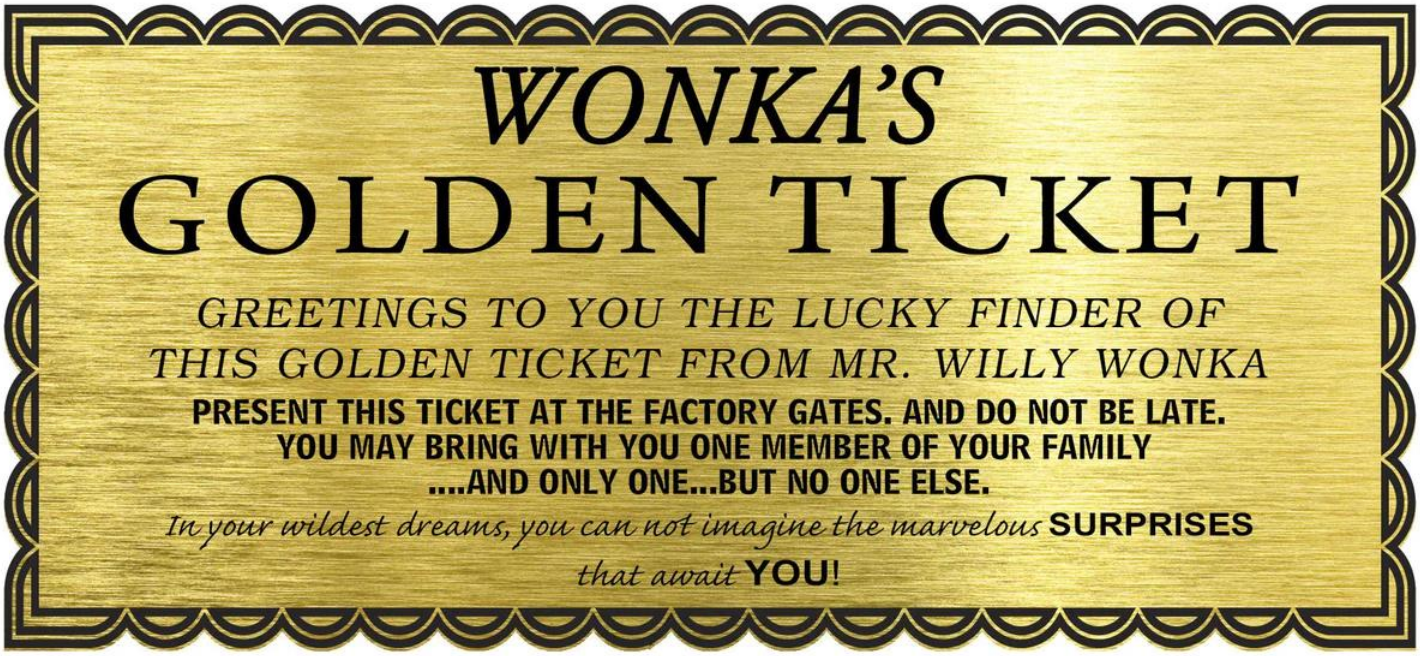
# CSC 1300 PROGRAM 3

FALL 2023

# WONKA'S CANDY INVENTORY




## IMPORTANT DATES:

**Assign date:** Wednesday, October 18, 2023

**Due Date:** Wednesday, November 1, 2023

You may submit up to 3 days late at 10 points off per day late. This means you can't submit after November 4, 2023, 11:59pm.

## CONCEPTS:

- Programmer-Defined Functions
- Pass-by-reference
- Arrays
- Parallel Arrays
- Header Files

## FILES TO TURN IN:

You will have **three files zipped together** in your submission that should be named as followed.

- jdoe42_prog3.h – this is your header file containing all necessary #includes, using namespace std, the constant variable, and all function prototypes.
- jdoe42_driver.cpp – this contains your main function.
- jdoe42_functions.cpp – this contains all your other functions.

## (replace jdoe42 with YOUR TTU username!!!)

Verify Submission: After submitting, make sure you can go back into assignments and see the file that you submitted. If you don't see it, you didn't submit. Then, make sure to download what you submitted to make sure your zip file isn't blank. If it is, you will get a zero for the submission.

## DESCRIPTION:

Willy Wonka is struggling to keep track of the prices of all his candies in the chocolate factory. He needs your help to determine the appropriate asking price for each candy. He wants to know the total cost of each type of candy, as well as the overall total cost. Additionally, he would like to know the profits of the candies and which candy is the most expensive in his inventory. Can you help him with this task?

## SPECIFICATIONS:

Write a menu-based program that will allow a user to do one of the five following tasks:

1. Add a piece of candy to inventory
2. Calculate totals for candy
3. Calculate profits for candy
4. Print the most expensive candy
5. Quit the program

Adding a piece of candy will involve:

- getting the type of candy (Wonka Bar, Everlasting Gobstoppers, Hair Toffee),
- the flavor of the candy,
- the cost of the materials to make the candy, and
- the number of Oompa Loompas it took to make the candy.
- Your program will then figure out the asking price of each piece of candy based on the following formula:

  *Asking price = number of Oompa Loompas * cost per Oompa Loompa + cost of materials*

  The cost of work per Oompa Loompa should be set to $7.5 and should be initialized in a constant variable in the header file.

While calculating totals, your program will display a table of each candy item and then the total for cost of materials and the total for asking price.

For calculating profits, your program should use the following formula:

*Profit = asking price - cost of materials*

You may NOT have any global variables – other than the cost of work per Oompa Loompa ($7.50).

You may NOT use concepts not introduced yet in class (e.g., structures/classes, pointers, etc.).

The user must be able to run the program as many times as they wish.

Validate all user inputs with loop.

Numbers entered by the user must not be negative so your program should tell the user if they entered a negative number and force them to enter a positive amount before the program will continue.

All dollar amounts must be printed to two numbers after the decimal point and dollar amounts for totals must be printed in the results with a dollar sign ($).

The output also must be in a neat, easy-to-read format. **Use stream manipulators.**

## REQUIRED FUNCTIONS

Your program will contain five functions in addition to the main function. You will define 5 arrays in the main function to store that candy data. Make sure you **do not write out of the bounds** of the five arrays in the program.

### MAIN FUNCTION

- The main function should have **five** arrays defined:
    - **candyType** – integer array with 50 elements that holds either a 1, 2 or 3 to represent what the candy type is (1= Wonka Bar, 2= Everlasting Gobstoppers, 3= Hair Toffee)
    - **candyFlavor** – an array of strings (**use the C++ string class!)** with 50 elements that holds an interesting flavor for each candy item.
    - **costMaterials** - float array with 50 elements that holds the cost of the materials to make the candy.
    - **numOompas** – int array with 50 elements that holds the number of Oompa Loompas it took to create the piece of candy.
    - **askingPrice** – float array with 50 elements that holds the asking price of the candy.

    You may have other arrays defined if needed.

- All numerical arrays should have all elements initialized to zero. You should also have **one integer variable** that holds the number of items in each array. The string array should be initialized to an empty string.

- You must let the user run the remainder of the program as many times as they want.

- Your program will output a menu of five choices:

    1. Add a piece of candy to inventory
    2. Calculate totals for candy
    3. Calculate profits for candy
    4. Print the most expensive candy
    5. Quit the program

- Let the user choose which option they want to do. (Validate the user's choice!)

- If the user chooses option 1 – you will call the **addCandy** function sending the number of items (passed-by-reference), candyType array, candyFlavor array, costMaterials array, numOompas array, and askingPrice array as arguments.
- For option 2 & 3 – Let the user choose what type of table to print from another menu of 4 options. Then, call the **calculateTotals/calculateProfit** function based on the user input.
- If the user chooses option 4 – call **getMaxPrice** function sending in the current number of candies in inventory, the askingPrice array and an int variable to store the max index passed by reference. Then, print the flavor, type and asking price of the most expensive candy (refer to the sample output).
- Lastly, if the user chooses option 5 – print a goodbye message and quit the program.

## addCandy FUNCTION

**Parameters**: an integer, holding the current number of candies passed by reference; int array for candyType; string array for candyFlavor; float array for costMaterials; int array for numOompas, and float array for askingPrice.

**Returns**: none

This function is called when the user wants to add a new candy to inventory. The function will accept the current number of items plus four arrays as parameters. The function will only add a new candy if there are less than 50 items in the arrays (to not go out of bounds).

To add a new candy, the program will have to ask the user for the candy type (by giving the user a menu and asking them to pick from one of the three options), ask for the flavor, ask for the cost of materials for this candy, and ask for the number of Oompa Loompas it took to create this candy. Validate the inputs! The user must not select an invalid type of candy or enter negative values for the cost or number of Oompa Loompas.

The program will add these values into the four respective arrays **at the correct index**.

Then, the program will call the **calculatePrice** function to populate the askingPrice array. Once all values are added, the program will print "A new candy has been added to the inventory!".

The function will then increment the number of items (passed by reference).

## calculateTotals FUNCTION

**Parameters**: an integer, representing what kind of table will be printed out (1= Wonka Bar, 2= Everlasting Gobstoppers, 3= Hair Toffee, 4 = all); an integer, holding the total number of candies in the arrays (current amount); and all five arrays defined in main.

**Returns**: none

This function will figure out what type of table the user wants based on the first integer that is passed into the function and **print out the respective table name**. For example, if the user wants to print out information on Wonka Bars, the header should read "Total Wonka Bars in Inventory:".

Then, the function will print out the table headers. There should be five columns and each column should have a header (words) at the top of the column explaining what that column is.

The values should be printed underneath the column headers. The values for total material cost and total asking price **must be calculated as running totals**.

Sample output for calculateTotals function:

```
Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 2


Choose what candy you want to get the total for:
        1.      Wonka Bars
        2.      Everlasting Gobstoppers
        3.      Hair Toffees
        4.      All candy

Choose 1-4: 4

Total Candy in Inventory:

TYPE                            FLAVOR      COST     NUM-OOMPAS    PRICE
-------------------------------------------------------------------------
Wonka Bar                       wasabi      3.40              2   18.40
Wonka Bar                  ghost pepper     3.00              3   25.50
Everlasting Gobstopper          booger      2.00              1    9.50
-------------------------------------------------------------------------

TOTALS:                              $      8.40              $   53.40
```

## calculatePrice FUNCTION

**Parameters**: a float, holding the cost of materials; an int holding the number of Oompa Loompas required to make the candy.

**Returns**: a float, holding the asking price of the candy.

This function calculates the asking price of a candy and then returns the asking price as a floating-point value.

The formula used for this function is:

*Asking price = number of Oompa Loompas * cost per Oompa Loompa + cost of materials*

where cost per Oompa Loompa set to **$7.5** (global constant variable).

## calculateProfit FUNCTION

**Parameters**: an integer, representing what kind of table will be printed out (1= Wonka Bar, 2= Everlasting Gobstoppers, 3= Hair Toffee, 4 = all); an integer, holding the total number of candies in the arrays (current amount); int array for candyType; float array for costMaterials; and float array for askingPrice

**Returns**: none

This function will figure out what type of table the user wants based on the first integer that is passed into the function and **print out the respective table name**. For example, if the user wants to print out information on Wonka Bars, the header should read "Total Profits for Wonka Bars in Inventory:"

Then you should print the candy types and their profits in a table as shown in the sample output. You will calculate the total profit as running total. Use the following formula to calculate the profits:

*Profit = asking price - cost of materials*

Sample output for calculateProfit function:

```
Total Profits for all candy in Inventory:

CANDY TYPE                        PROFIT
---------------------------------------
Wonka Bar                          15.00
Everlasting Gobstopper              7.50
Wonka Bar                          22.50
---------------------------------------

TOTAL Profit:                $     45.00
```

## getMaxPrice FUNCTION

**Parameters**: an integer, holding the total number of candies in the arrays (current amount); float array for askingPrice; and an integer variable, **passed by reference** that would hold the index for the most expensive candy.

**Returns**: float holding the asking price for the most expensive candy.

Use loops to find out the highest asking price for all the candies (store it in a float variable) and the index number of that candy (with the maximum asking price). Return the float containing the highest asking price.

## CODE READABILITY:

- Provide a comment block at the top of your source file that contains the source file name, your name, creation date, and purpose of the program. Put labels before this data like:
  Filename:
  Author:
  Creation Date:
  Purpose:

- Place a comment above each function telling the function name & purpose.

- Name your variables where they describe what they hold. Accumulator variables should have the word "total" or "sum" in them.

- Indent your code properly and consistently.

## SAMPLE OUTPUT:

```
************************************************
*  WELCOME TO WILLY WONKA'S CHOCOLATE FACTORY!!  *
************************************************

Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 9

You must enter 1-5!
Choose again: 1

Choose a candy type:
        1. Wonka Bar
        2. Everlasting Gobstoppers
        3. Hair Toffee

Enter choice: 6

You must enter 1, 2, or 3!
Enter choice: -3

You must enter 1, 2, or 3!
Enter choice: 1

What's the flavor of your candy? Wasabi
How much did the material cost? $13.4
How many Oompa Loompas did it take to make the candy? -2

Error!! You cannot have a negative number of Oompa Loompas!
How many Oompa Loompas did it take to make the candy? 2

A new candy has been added to the inventory!


Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 1

Choose a candy type:
        1. Wonka Bar
        2. Everlasting Gobstoppers
        3. Hair Toffee

Enter choice: 2
```

```
What's the flavor of your candy? booger
How much did the material cost? 3
How many Oompa Loompas did it take to make the candy? 1

A new candy has been added to the inventory!


Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 1

Choose a candy type:
        1. Wonka Bar
        2. Everlasting Gobstoppers
        3. Hair Toffee

Enter choice: 1

What's the flavor of your candy? ghost pepper
How much did the material cost? 4
How many Oompa Loompas did it take to make the candy? 3

A new candy has been added to the inventory!


Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 2


Choose what candy you want to get the total for:
        1.      Wonka Bars
        2.      Everlasting Gobstoppers
        3.      Hair Toffees
        4.      All candy

Choose 1-4: 1


Total Wonka Bars in Inventory:

TYPE                            FLAVOR      COST     NUM-OOMPAS   PRICE
-------------------------------------------------------------------------
Wonka Bar                       Wasabi     13.40              2   28.40
Wonka Bar                  ghost pepper     4.00              3   26.50
-------------------------------------------------------------------------
```

```
TOTALS:                                  $     17.40              $   54.90



Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 2


Choose what candy you want to get the total for:
        1.      Wonka Bars
        2.      Everlasting Gobstoppers
        3.      Hair Toffees
        4.      All candy

Choose 1-4: 4


Total Candy in Inventory:

TYPE                                FLAVOR      COST     NUM-OOMPAS   PRICE
-------------------------------------------------------------------------------
Wonka Bar                           Wasabi     13.40              2   28.40
Everlasting Gobstopper              booger      3.00              1   10.50
Wonka Bar                     ghost pepper      4.00              3   26.50
-------------------------------------------------------------------------------

TOTALS:                                  $     20.40              $   65.40



Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 3


Choose what candy you want to get the profit for:
        1.      Wonka Bars
        2.      Everlasting Gobstoppers
        3.      Hair Toffees
        4.      All candy

Choose 1-4: 2


Total Profits for Everlasting Gobstopper in Inventory:
```

```
CANDY TYPE                        PROFIT
---------------------------------------
Everlasting Gobstopper             7.50
---------------------------------------

TOTAL Profit:               $      7.50



Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 3


Choose what candy you want to get the profit for:
        1.      Wonka Bars
        2.      Everlasting Gobstoppers
        3.      Hair Toffees
        4.      All candy

Choose 1-4: 4


Total Profits for all candy in Inventory:

CANDY TYPE                        PROFIT
---------------------------------------
Wonka Bar                          15.00
Everlasting Gobstopper              7.50
Wonka Bar                          22.50
---------------------------------------

TOTAL Profit:               $     45.00



Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 3


Choose what candy you want to get the profit for:
        1.      Wonka Bars
        2.      Everlasting Gobstoppers
        3.      Hair Toffees
        4.      All candy
```

```
Choose 1-4: 1


Total Profits for Wonka Bar in Inventory:

CANDY TYPE                        PROFIT
---------------------------------------
Wonka Bar                          15.00
Wonka Bar                          22.50
---------------------------------------

TOTAL Profit:                $     37.50



Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 4

Candy with the highest price is the Wasabi flavored Wonka Bar for $28.40


Wonka's candy management program:
        1.      Add a piece of candy to inventory
        2.      Calculate totals for candy
        3.      Calculate profits for candy
        4.      Print candy with max price
        5.      Quit the program

Choose 1-5: 5

        GOODBYE!!
```