# CSC134 Project

| | |
|---|---|
| **Project Number** | Project 4 |
| **Project Name** | Postal Barcode |
| **Project Filename** | Firstname_Lastname_Project4 (example: *Jessica_Smith_Project4*) |
| **Chapter Review** | Chapter 6<br>Use only the concepts covered in Chapters 1 - 6 to complete this project. Failure to do so may result in a 0 for the assignment.<br>Students are expected to complete the practice programs before starting the project. |
| **Points** | 50 |
| **Assistance** | Instructors and teaching assistants have been available to assist with **all** practice work. Students are now expected to complete this project **without** assistance from others (this includes receiving assistance from individuals inside or outside of CPCC). Students should consider projects as **non-proctored exams**. Please review the academic integrity policy on your syllabus.<br>**Please note**: students can continue to receive assistance with practice work up to 5:00 p.m. on the due date (review the late period on the syllabus). |
| **Project Description** | For this part of the project, you will break up your program into manageable pieces. Instead of having one long main function that contains all the statements necessary to solve the problem given in previous projects, you must write several small functions that each solve a specific part of the problem. These small functions should then be executed in the correct order to implement a complete solution. The named constants you used for the postage rate calculations should now be made global so you won't have to pass them to any of your functions.<br>Use the *Divide and Conquer* approach described in **Section 6.1** of your textbook to modularize your program. Think about all the different tasks your program is doing and make a list. Make sure each task doesn't include multiple tasks. If it does, break it into subtasks. Once you have your list of tasks and subtasks, you'll know what functions you'll need to write. For this project, your list MUST include the following:<br>• a function that displays the menu<br>• a function that calculates and returns the checksum digit<br>• a function that accepts a single digit and returns the encoded value<br>• a function that calculates and returns the postage for a single mailing<br>Make sure your program uses the values returned from your functions. Any functions that need input to perform a task will have to accept parameters. **Global variables are prohibited**.<br>You are strongly encouraged to use the *Stubs and Drivers* approach the author of your textbook talks about in **Section 6.16** when writing your functions. Write one function at a time thoroughly testing one before starting another. This will isolate any errors to just one block of code making your program much easier to debug. |

**Instructions:**

1. Use the *Divide and Conquer* approach described in **Section 6.1** of your textbook to modularize your program.

2. Write a function that displays the menu.

3. Write a function that calculates and returns the checksum digit.

4. Write a function that accepts a single digit and returns the encoded value.

5. Write a function that calculates and returns the postage for a single mailing.

6. Make all the postage rate named constants global.

7. Save your **.cpp** file using the Project**4**_Firstname_Lastname naming format. Upload in Moodle.

**Bonus Pts (5pts).** Add functionality so that if a user selects menu option 2, your program will also create **one** text file containing mailing labels (formatted exactly like the **Sample Output**) for **every** mailing in the ***Mailings.txt*** file. To earn the bonus points, you must write and use a function that writes **one** mailing label to the text file each time it is called. *HINT: Research Chapter 12.* The total cost of all the mailings should also be written to the file.

## Sample Output

```
Welcome to the Mailing Label Printing System

1 – Single Mailing
2 – Multiple Mailings
3 -- Quit

Enter your choice:  1

Enter name : Larry Smith
Enter street address: 122 Main Street
Enter city: Charlotte
Enter state: NC
Enter zip code: 23499
Enter 1 for letter, 2 for envelope, 3 for parcel: 1
Enter weight in ounces: 2.1

**************************$0.93

Larry Smith
122 Main Street
Charlotte     NC  23499

|::|:|::||::|::||:|::|:|::::||:|


1 – Single Mailing
2 – Multiple Mailings
3 -- Quit

Enter your choice:  2

**************$2.94

Ann Perkins
217 Lowell Drive
Pawnee  IN   47408

|:|::||:::|:|::|||:::|::|:|:::||


**************$1.42

Ron Swanson
3657 White Bridge Road
Eagleton  IN   47320

|:|::||:::|::||:::|:|||::::|::||
```

```
**************$0.71

Andy Dwyer
789 Cherry Road
Wamapoke  IN   48034

|:|::||::|:||:::::||::|::|:::|||


**************$0.49

Ben Wyatt
1513 Plymouth Street
Pawnee  IN   47407

|:|::||:::|:|::|||:::|:::||::|:|


**************$2.54

Chris Traeger
8900 Rapport Street
Eagleton  IN   47322

|:|::||:::|::||:::|:|::|:|::|:||


Total cost of all the postage is $8.10

1 – Single Mailing
2 – Multiple Mailings
3 -- Quit

Enter your choice:  3

Thank you.  Closing program.
```

## GradingScore Card: 50

| OBJECTIVE | DESCRIPTION | POINTS |
|---|---|---|
| Program is modular | Program has been completely modularized using the Divide and Conquer approach. All functions perform a single task and correct output is produced for all input. | 10 |
| | All four of the required functions have been written as specified and are called correctly. | 20 |
| | Functions have self-describing names. | 5 |
| | Function prototypes exist for all functions. | 5 |
| Global Named Constants | Named constants that store postage rates are used in all calculations involving postage computation and are not passed as an arguments to any of the functions. No global variables are used. | 5 |
| Documentation and Programming Style | • Multi-line comment present at the beginning of the program that contains: your **Name**, **Date** and **Purpose** which fully describes what your program does.<br>• Adequate descriptive comments used throughout program. There must be at least four comments to receive credit for this step.<br>• Proper indentation and spacing used throughout program.<br>• Proper naming conventions used throughout program. | 5 |
| **TOTAL** | | **50** |
| **Bonus Points** | Functionality was added so that if a user selects menu option 2, **one** text file is created that contains mailing labels (formatted exactly like the **Sample Output**) for **<u>every</u>** mailing in the ***Mailings.txt*** file along with the total cost of all the mailings. A function that writes **<u>one</u>** mailing label to the text file each time it is called was written and used. | **+5** |
| **Point Deduction** | Program does not compile without errors. | **-20** |