CS 161B: Programming and Problem Solving II

Assignment 2: Encoder

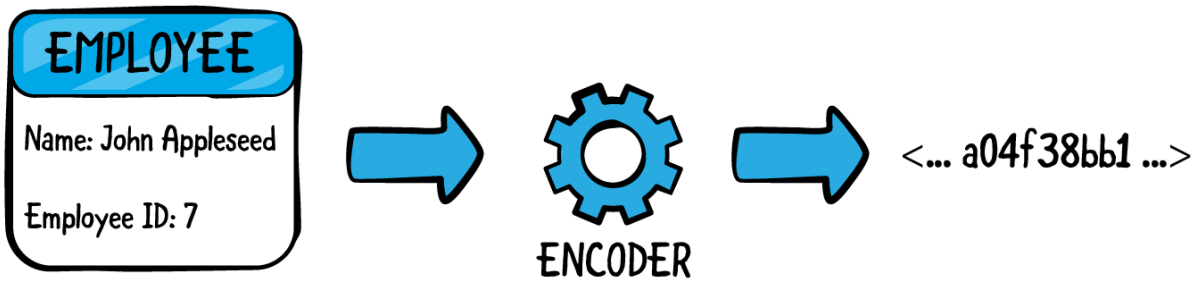
Academic Integrity

You may NOT, under any circumstances, begin a programming assignment by looking for completed code on StackOverflow or Chegg or any such website, which you can claim as your own. Please check out the Student Code of Conduct at PCC.

The only way to learn to code is to do it yourself. The assignments will be built from examples during the lectures, so ask for clarification during class if something seems confusing. If you start with code from another source and just change the variable names or other content to make it look original, you will receive a zero on the assignment.

I may ask you to explain your assignment verbally. If you cannot satisfactorily explain what your code does, and answer questions about why you wrote it in a particular way, then you should also expect a zero.

Have you ever encountered a text encoder? Now you can write one yourself. This is the first term we are using D2L across the University as our learning management system. It will take some adjustment to get used to it. One thing I have found working with D2L is that it changes your file names when we download submissions. There is “some” logic to the file name it creates, and you are going to write a program to encode the fileNames!!



Purpose

Your job is to design and create a program to allow for PCC to encode and build the file name for D2L. You should ask the user at the beginning if they are building the name of the file (encoding it) or if the program wants to quit. Then, the user will need to enter in information to allow the program to build the encoded file name or quit.

After completing this assignment you will be able to:

- Use the cstring library for all appropriate string functions like `strncpy(), strcat()` etc.
- Use arrays of characters (char arrays) or cstrings.
- Apply the concept of overloaded functions, and functions with value and reference parameters.

Task

- ❏ Open the Algorithmic Design Document, make a copy, and follow the steps to create your algorithm.
- ❏ You must express your algorithm as **pseudocode,** and you must have the following functions in your code.
- ❏ Introduce the program to the user - `void welcome()` function.
- ❏ Allow the user to select if they want to encode (create) a filename or quit the program.
    - ❏ Write a function `char displayMenu()` that displays the menu and reads the option into a char variable and returns it.
    - ❏ Prompt and read in from the user what they want to do today. All options must be validated in the function.
- ❏ If the Encode option is selected, call the `encode()` function and send a char array to the function. The function should call other functions and fill this char array which should be printed in `main` before the menu loops again.
- ❏ The `encode` function: This should be a void function takes 1 char array `encodeFileName` and does the following:
    - ❏ Declare appropriate local variables.
    - ❏ Call the `readInput()` function to read student's names and lateFlag.
    - ❏ Call the `readInput()` function to read Student ID and filename.
    - ❏ Call the `readTime()` function to read the submitted time.
    - ❏ Fill the `encodeFileName` array based on the six pieces of information adding underscores between.

- Start with the student's names, if the assignment is late, put LATE in the filename and nothing otherwise, add the parsed Student ID, add the time, without the colon, and then finally add the filename.
- For example: If late: smith_sue_LATE_5587_1824_prog2.c
- If it is not late: smith_sue_5587_1824_prog2.c
- Use **`strncpy() and strcat()`** functions - **refer to Sections 10.4 and 10.5 C-String library functions** in your zyBooks. Look at my example code file.

- `readInput(char fName[], char lName[], bool &lateFlag) - overloaded function`
  - This void function should take 2 char arrays and 1 bool variable by reference.
  - It reads the student's first name, last name and if the assignment is late or not. Student last name (e.g. Smith) **(char array).** Student first name (e.g. Sue) **(char array)**
  - Make sure all character data for the student's name is lowercase - you may write a function to convert a cstring to all lowercase.
  - If the assignment was late or not (e.g., Y or N) **(bool var)**. Must do data validation for this.
  - ***Hint: Remember an overloaded function has the same function name but different types of parameters or the number of parameters are different.***
- `readInput(char parsedID[], char fileName[]) - overloaded function`
  - This void function should take 1 char array for the parsed 4 digit Student ID that will be returned and another char array for the filename.
  - The Student ID (e.g., 977-15-5587) **(char array)** is a local variable. Use the strncpy() function to copy from position 7 till the end of the string. **`strncpy(parsedID, stdID + 7, 4)`** will copy 4 characters from the 7th position in the stdID string to the parsedID string. Check out this example code file.
  - The name of the file (e.g., prog2.cpp) **(char array)**
- `readTime(char strTime[])`
  - This is a void function that should read the time from the user as 2 integers and return one char array by reference (that is the strTime array).
  - The time read from the user will be in military time (e.g.. 18:24 for 6:24pm) **(2 ints, one for hour and one for mins).** Check out this example code file to extract the numbers and convert them to a cstring. This example is not quite sophisticated. You can do a little more by checking the hours are between 0 and 24 and the minutes are between 0 and 60.
  - This function must do data validation for the numbers and the HH:MM format.
- main()

- ❏ Declare appropriate local variables.
- ❏ In an appropriate loop call the functions and output the encoded file name.
- ❏ Repeat the menu in the loop until the user chooses to quit.
- ❏ `main()` should do nothing more than the above steps.

- ❏ **You may not use any temporary arrays to help you solve this problem. (But you may declare as many simple variables as you like, such as ints.) You also may not use any other data structures or complex types such as strings, or other data structures such as Vector. Use only the concepts and functions we have learned so far.**
- ❏ Your program must have function prototypes. Place the prototypes for your functions globally, after your #includes. All functions must be implemented after `main()`.
- ❏ Try not to have any redundant code (repeated code) in your program. That is the purpose of functions.
- ❏ Print a goodbye message.
- ❏ Please follow ALL the instructions to submit your file in the Criteria for Success section below.

Criteria for Success

- ❏ Test your program using the following sample runs, making sure you get the same output when using the given inputs (in **`blue`**).
- ❏ Make sure to test your program with different sets of data.

```
Welcome to my fileName encoding program!!

Please pick an option below:
(e)Encode a file name
(q)quit
>>e
This program will ask you a few questions and generate an
encoded fileName based on your answers.

Enter your last name: Iyer

Enter your first name: GD

Was your assignment Late (y/n)? Y

Enter your Student-ID (format: 222-22-2222): 234-05-4556

Enter the file name: a05.cpp
```

```
Enter the time submitted (military time - ex: 18:24 for 6:24pm):
13:45

Your encoded file name is: iyer_gd_LATE_4556_1345_a05.cpp

Please pick an option below:
(e)Encode a file name
(q)quit
>>b
Invalid option! Please try again!!
Please pick an option below:
(e)Encode a file name
(q)quit
>>q

Thank you for using my fileName generator!
```

- ❏ Complete zyBooks section **Chapter 10: CS 161B Char Arrays** activities.
- ❏ You may use any IDE to write your code. If your Transfer University is PSU or plan to continue with CS courses here at PCC, check out the Special Note For PSU Transfer section in the Syllabus.
- ❏ Please bookmark the PCC Linux and Vim Manual - this will become a frequently used reference!
- ❏ Complete all sections of your Algorithmic Design Document and include the **pseudocode** in part d of the design document.
- ❏ **Follow these Coding Construct Requirements:**
  - ❏ Must have all the functions mentioned above.
  - ❏ Your program must have function prototypes. Place the prototypes for your functions globally, after your #includes. **All functions must be implemented after main()**.
  - ❏ Must do data validation where appropriate and where required in the functions above.
  - ❏ Must read the time as 2 ints and the convert to a char array.
  - ❏ Print a welcome and goodbye message.
- ❏ Please open and compare your work with the grading rubric before submitting.
- ❏ Remember to follow all style guidelines.
- ❏ Download your Algorithmic Design Document as a PDF (File -> Download -> PDF), rename it to `a02.pdf`, and upload it to the D2L assignment by **Wednesday**.
- ❏ Upload your `a02.cpp` C++ source file to the D2L assignment by **Sunday**.

- ❏ Do your own work. Consult the syllabus for more information about academic integrity.

Additional Support

- ❏ Use **`strncpy()`** and **`strcat()`** - look at my example code file.
- ❏ Check out this example code file to extract the numbers and convert them to a cstring.
- ❏ Post a question for the instructor in the Ask Questions! area of the Course Lobby.