Lab4

Due Sep 16, 2020 by 11:59pm **Points** 40 **Submitting** a website url

Available until Sep 17, 2020 at 12:05am

This assignment was locked Sep 17, 2020 at 12:05am.

Lab4:

DUE: Wednesday, September 16th @ 11:59 PM. You must use the CODE plug in to submit your solution. In order to be submitted, the code must compile with no errors or warnings and must pass the cpplint.py style checker with no errors.

Background: This lab will check your understanding of a few concepts you have learned about over the past few weeks:

- How to use unordered_maps
- How to pass parameters to functions in a way that lets them be updated
- How to continuously prompt a user until their input is valid

Procedure: Download the supplied starter code **lab4.cpp (https://miamioh.instructure.com/courses/129645/files/17175189/download?wrap=1)** **(https://miamioh.instructure.com/courses/129645/files/17175189/download?download_frd=1)** and modify it according to these steps and the comments in the starter code.

1. In the main function, create an appropriate unordered_map that will map a std::string to an int **[6 points]**
2. Declare instances of the map called weekMap and monthMap **[3 points]**
3. Invoke the createWeekMap function with an appropriate argument to update the weekMap unordered_map **[1 point]**
4. Invoke the createMonthMap function with an appropriate argument to update the monthMap unordered_map **[1 point]**
5. Until the user enters a valid month and day, loop over the following **[8 points]**
    - Prompt the user to enter a month (e.g., *April*) and a weekday (e.g., *Monday*). See the code comments for the required format. **[3 points]**
        - **NOTE: the month and weekday should be case-sensitive (as per the starter code). For example, April is a valid month, but april would be invalid**

- If the user enters an incorrect month, inform them with a message to standard error. See the code comments for the required format. **[2 points]**
- You should use an appropriate method acting on your monthMap instance to determine if the input is valid. **[2 points]**
- If the user enters a correct month, check for a correct day. If the user enters an incorrect day, inform them with a message to standard error. See the code comments for the required format. **[2 points]**
- You should use an appropriate method acting on your weekMap instance to determine if the input is valid. **[2 points]**

6. When both the month and week are valid, write the mapping of their month and day to the appropriate integer values. See the code comments for the required format. **[2 points]**
7. Add code to createWeekMap that will create a map of the days of the week. **[2 points]**
   - The method should accept arguments that allow for the changes in the weekMap unordered_map to be passed back to main. **[2 points]**
8. Add code to createMonthMap that will create a map of the months of the year. **[2 points]**
   - The method should accept arguments that allow for the changes in the monthMap unordered_map to be passed back to main. **[2 points]**

Example Output:

Below is a sample output when running the program in the terminal. The pink represents the user input and blue represents program output

Example 1:

```
[terminal_prompt]$./lab4
Enter a month and weekday: April Apple
The day Apple is not valid
Enter a month and weekday: April Wednesday
April, Wednesday => 4, 4
```

Example 2:

```
[terminal_prompt]$./lab4
Enter a month and weekday: Apple Wednesday
The month Apple is not valid
Enter a month and weekday: April Wednesday
April, Wednesday => 4, 4
```

Example 3:

```
[terminal_prompt]$./lab4
Enter a month and weekday: May Friday
May, Friday => 5, 6
```

Lab 4 Rubric

Criteria	Ratings		Pts
Create an appropriate unordered_map that will map a std::string to an int	**6 pts** **Full Marks**	**0 pts** **No Marks**	6 pts
Declare instances of the map called weekMap and monthMap	**3 pts** **Full Marks**	**0 pts** **No Marks**	3 pts
Invoke the createWeekMap function with an appropriate argument to update the weekMap unordered_map	**1 pts** **Full Marks**	**0 pts** **No Marks**	1 pts
Invoke the createMonthMap function with an appropriate argument to update the monthMap unordered_map	**1 pts** **Full Marks**	**0 pts** **No Marks**	1 pts
Until the user enters a valid month and day, loop over the following	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
Prompt the user to enter a month (e.g., April) and a weekday (e.g., Monday). See the code comments for the required format.	**3 pts** **Full Marks**	**0 pts** **No Marks**	3 pts
If the user enters an incorrect month, inform them with a message to standard error. See the code comments for the required format.	**2 pts** **Full Marks**	**0 pts** **No Marks**	2 pts
You should use an appropriate method acting on your monthMap instance to determine if the input is valid.	**2 pts** **Full Marks**	**0 pts** **No Marks**	2 pts
If the user enters a correct month, check for a correct day. If the user enters an incorrect day, inform them with a message to standard error. See the code	**2 pts**	**0 pts**	2 pts

Criteria	Ratings		Pts
comments for the required format.	**Full Marks**	**No Marks**	
You should use an appropriate method acting on your weekMap instance to determine if the input is valid.	**2 pts Full Marks**	**0 pts No Marks**	2 pts
When both the month and week are valid, write the mapping of their month and day to the appropriate integer values. See the code comments for the required format.	**2 pts Full Marks**	**0 pts No Marks**	2 pts
Add code to createWeekMap that will create a map of the days of the week.	**2 pts Full Marks**	**0 pts No Marks**	2 pts
The method should accept arguments that allow for the changes in the weekMap unordered_map to be passed back to main.	**2 pts Full Marks**	**0 pts No Marks**	2 pts
Add code to createMonthMap that will create a map of the months of the year.	**2 pts Full Marks**	**0 pts No Marks**	2 pts
The method should accept arguments that allow for the changes in the monthMap unordered_map to be passed back to main.	**2 pts Full Marks**	**0 pts No Marks**	2 pts
			Total Points: 40