Lab16a will be graded separately from Lab15a

This lab exercise is a ***successor*** of Lab15a: If you have not completed Lab15a, you should complete that assignment **first**, submit the solution to *Moodle* and demonstrate the solution during class.
When you begin working on Lab16a, **first** make a copy of your final Lab15a solution and **rename** the **`Lab15a.cpp`** source code to **`Lab16a.cpp`**. The two assignments will be graded separately and must be *demonstrated* separately.

Due Date

You must submit the source code file for the solution to this lab exercise to Moodle by

Thursday, August 7, 2025

in order to receive full credit for this work. You must also demonstrate the solution to the instructor during class, at the earliest opportunity.

Summary of Lab15a

In Lab15a, you were given code for the **Employee** class (**`Employee.h`** and **`Employee.cpp`**), and the class specification for a class named **ProductionWorker** (**`ProductionWorker.h`**). The **ProductionWorker** class is derived from the **Employee** class: that is, the **ProductionWorker** class is a subclass of the **Employee** class.

Special Note

The Lab15a assignment provided the source code files (**`Employee.h`**, **`Employee.cpp`**, and **`ProductionWorker.h`**) with the requirement that the student *must not* modify those files. This assignment provides the same files again, as a reminder that in this assignment you do need to modify these files.

Programming Assignment: Use Exceptions for Error Reporting

1. Make a copy of your final Lab15a solution and **rename** the **`Lab15a.cpp`** source code to **`Lab16a.cpp`**. (The two assignments will be graded separately.)
2. Modify the **Employee** and **ProductionWorker** classes to define *exception classes* and use them in the program to report errors in the user input.

Employee Class

Modify the **Employee** class:

- Add an exception class: **InvalidHireDate**
- Add code to the **Employee** class to check if the hire date **string** object fits the **MM/DD/YYYY** numeric format. One easy way to accomplish this is to use the "square brackets" operator (**`[ ]`**) to access individual characters in the

  ```
  string hireDate
  ```

  member variable in the **Employee** object:

1. The **`hireDate`** string must have a length of **`10`**.
2. The characters at index **`2`** and index **`5`** must be a forward-slash character ('**`/`**').
3. The characters at index **`0`**, **`1`**, **`3`**, **`4`**, **`6`**, **`7`**, **`8`**, and **`9`** must be in the range of **`0..9`**. Refer to the **`isdigit()`** function in the **`cctype`** function library . You may need to add a **`#include <cctype>`** statement to your program. (Refer to Chapter 10 of the textbook, or the **cplusplus.com** web-site.)

ProductionWorker Class

Modify the **ProductionWorker** class:

- Add two exception classes: **InvalidShift**, and **InvalidPayRate**.
- Add two new test functions:

  `testShift(int shift)` and
  `testPayRate(double rate)`

  These functions must test the validity of the calling parameter, and throw the appropriate exception if the parameter is incorrect.

- Enhance the **static** function for creating a new **ProductionWorker** object:

  `static ProductionWorker *createNewProductionWorker();`

  Modify the code so that the **ProductionWorker** object is dynamically created inside a **`try`** block. After the **`try`** block there should be **`catch`** blocks to handle the three possible types of exception: **InvalidHireDate**, **InvalidShift**, and **InvalidPayRate**.
  If an error occurred, ask the user for new input data and try again. Repeat the process until a **ProductionWorker** object is successfully created.

Test the Program

Test the program with some valid input values and some invalid values.
(Refer to the **Sample Input / Output** on the following pages.)

Demonstrate the Program During Class

As always, you must demonstrate your working solution **during class** to get credit for the assignment.

(Continued on the next page.)

Sample Input / Output

In the sample input/output session that follows, the **bold** text is what the user entered. In actuality, all text (both input and output) will be displayed in the same font.

Sample Input / Output Session

```
Enter command (or 'h' for help): h
Supported commands:
        c               create a new ProductionWorker object.
        h               print help text.
        p               print ProductionWorker information.
        q               quit (end the program).

Enter command (or 'h' for help): c
Enter name of new employee: George Washington
Enter hire date of new employee: 04/30/1789
Enter shift for new employee (1=day, 2=night): 1
Enter hourly pay rate for new employee: 35.43
Enter command (or 'h' for help): p
Name: George Washington
Employee number: 1
Hire date: 04/30/1789
Shift: Day
Shift number: 1
Pay rate: 35.43
Enter command (or 'h' for help): c
Enter name of new employee: John Adams
Enter hire date of new employee: 3/4/1797
Enter shift for new employee (1=day, 2=night): 1
Enter hourly pay rate for new employee: 50.33
Error: Invalid hire date [3/4/1797]:  Hire date must be MM/DD/YYYY
format.
Enter name of new employee: Thomas Jefferson
Enter hire date of new employee: 03/04/1801
Enter shift for new employee (1=day, 2=night): 1
Enter hourly pay rate for new employee: 64.53
Enter command (or 'h' for help): p
Name: Thomas Jefferson
Employee number: 3
Hire date: 03/04/1801
Shift: Day
Shift number: 1
Pay rate: 64.53
```

Sample Input / Output Session

```
Enter command (or 'h' for help): c
Enter name of new employee: James Madison
Enter hire date of new employee: 03/04/1809
Enter shift for new employee (1=day, 2=night): 2
Enter hourly pay rate for new employee: -88.44
Error: Invalid pay rate:  -88.44
Enter name of new employee: James Madison
Enter hire date of new employee: 03/04/1809
Enter shift for new employee (1=day, 2=night): 2
Enter hourly pay rate for new employee: 88.44
Enter command (or 'h' for help): p
Name: James Madison
Employee number: 5
Hire date: 03/04/1809
Shift: Night
Shift number: 2
Pay rate: 88.44
Enter command (or 'h' for help): c
Enter name of new employee: James Monroe
Enter hire date of new employee: 03/04/1817
Enter shift for new employee (1=day, 2=night): 3
Enter hourly pay rate for new employee: 3.44
Error: Invalid shift number:  3
Enter name of new employee: James Monroe
Enter hire date of new employee: 03/04/1817
Enter shift for new employee (1=day, 2=night): 1
Enter hourly pay rate for new employee: 43.44
Enter command (or 'h' for help): p
Name: James Monroe
Employee number: 7
Hire date: 03/04/1817
Shift: Day
Shift number: 1
Pay rate: 43.44
Enter command (or 'h' for help): q
```