Due Date:

You must demonstrate the solution to this lab exercise to the instructor by **Friday, March 29, 2019**, in order to receive full credit for this work.

Employee class – provided as "starter code"

The "starter code" for this lab is a class named **Employee**. Objects of the **Employee** class represent employees of a company. This class is carefully disigned to generate *sequential* values for the **employeeNumber** member variable: the first **Employee** object created receives an **employeeNumber** value of 1, the next Employee object created receives an **employeeNumber** value of 2, etc. (How this works was discussed in class, as well as in Chapter 14 of the Gaddis textbook.)

Your code for this lab exercise must use the **Employee** class, but you may **not** modify the **Employee** class for this lab exercise.

ProductionWorker Class

Write a class named **ProductionWorker** that is a sub-class of the **Employee** class. That is, the *class specification* file for the **ProductionWorker** class should be similar to the following example:

ProductionWorker.h

```
// Specification file for the ProductionWorker Class
#ifndef PRODUCTION_WORKER_H
#define PRODUCTION_WORKER_H
#include <iostream>
#include <iomanip>
#include <string>
#include "Employee.h"

using namespace std;

class ProductionWorker : public Employee
{
private:
      int shift;         // The worker's shift
      double payRate;    // The worker's hourly pay rate

public:
      // Default constructor
      ProductionWorker() : Employee() {
            shift = 0; payRate = 0.0;
      }


      // Constructor
      ProductionWorker(string aName, string aDate, int aShift, double aPayRate)
            : Employee(aName, aDate) {
            shift = aShift; payRate = aPayRate;
      }
```

ProductionWorker.h

```
        // Mutators
        void setShift(int s);
        void setPayRate(double r);
        static ProductionWorker *createNewProductionWorker();

        // Accessors
        int getShiftNumber() const;
        string getShiftName() const;
        double getPayRate() const;
        void printWorkerData() const;

        //******************************************************
        // The displayInfo function displays a production      *
        // worker's employment information.                    *
        //******************************************************
        void displayInfo(ProductionWorker e);
};

#endif
```

Most of the actual code for the ProductionWorker class should be in a file named **ProductionWorker.cpp**.

The **ProductionWorker** class should have member variables to hold the following information:

- Shift (an integer): a value of **1** means “day shift”, and **2** means “night shift”.
- Hourly pay rate (a double):

Write one or more constructors and the appropriate accessor and mutator functions for the **ProductionWorker** class.

Demonstrate the **Employee** and **ProductionWorker** classes by writing a “main” program that uses a **ProductionWorker** object. The “main” function should include a “command loop” similar to those which we have used in previous labs. (Feel free to *re-use* portions of your code from earlier labs.) The command loop must support the commands described in the following “help text”:

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

Sample Interactive Session

In the sample data on the next page, what the user types is shown in **bold**. In actuality, what the user types would appear as the same text format as the rest of the output.

```
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: April 30, 1789
Enter shift for new employee (1=day, 2=night): 2
Enter hourly pay rate for new employee: 123.45
Enter command (or 'h' for help): p
Name: George Washington
Employee number: 1
Hire date: April 30, 1789
Shift: Night
Shift number: 2
Pay rate: 123.45
Enter command (or 'h' for help): c
Enter name of new employee: John Adams
Enter hire date of new employee: March 4, 1797
Enter shift for new employee (1=day, 2=night): 2
Enter hourly pay rate for new employee: 543.21
Enter command (or 'h' for help): p
Name: John Adams
Employee number: 2
Hire date: March 4, 1797
Shift: Night
Shift number: 2
Pay rate: 543.21
Enter command (or 'h' for help): c
Enter name of new employee: Thomas Jefferson
Enter hire date of new employee: March 4, 1801
Enter shift for new employee (1=day, 2=night): 1
Enter hourly pay rate for new employee: 567.89
Enter command (or 'h' for help): p
Name: Thomas Jefferson
Employee number: 3
Hire date: March 4, 1801
Shift: Day
Shift number: 1
Pay rate: 567.89
Enter command (or 'h' for help): q
```