Objective:

Work with classes, strings, separate compilation, and operator overloading.

Design a class called `Appointment` that stores the properties of a calendar appointment. Properties include the appointment title, date (**year**, **month**, **day)**, starting **time (military time)**, and duration. Your class must check for invalid values for year, month, day, time, and duration. Your class must include the following getters and setters for all member data (**function names and prototypes must match exactly**):

- `Appointment(); Default constructor`
  It should initialize the title to "N/A". The date should be initialized to 1, 1, and 1. The time should be initialized to 0, and duration to 1.
- `Appointment(string appData);`
  Add an appointment from a string in the format `"title|year|month|day|time|duration"`. Time is given in standard time and duration is given in minutes. Leading and trailing spaces must be removed.
  Example: `" Meeting with Bob | 2019 |4 |29 |8:30 aM | 15 "`

- `string getTitle();`
- `int getYear();`
- `int getMonth();`
- `int getDay();`
- `int getTime(); //military time`
- `int getDuration();`
- `string getDate(); //return a date in the format 2019-02-13`
- `string getStandardTime();`
- `void setTitle(string newTitle);`
- `void setYear(int newYear);`
- `void setMonth(int newMonth);`
- `void setDay(int newDay);`
- `void setTime(int newTime); //military time`
- `void setDuration(int newDuration);`
- `void setDate(int year, int month, int day);`

In addition, the class must include the following helper functions:

- `string militaryToStandard(int time);`
  Converts time from military (`1830`) to standard time (`"6:30PM"`)
- `int standardToMilitary(string time);`
  Converts standard time (`"8:30PM"`) to military time (`2030`). It should handle lower- and upper-case letters.
  return the time in standard time format (`2030`).
- `bool operator ==(const Appointment &first, const Appointment &second);`

Write a main program to test all the functions above. Make sure you check for invalid values. You can verify the correctness of your code by checking testing output on Travis-CI (more on this in class).

Separate Compilation:
First, test your program using a single file "appointment_main.cc". Divide your project into three files:

- `appointment.h`
  - Contains the class definition
- `appointment.cc`
  - Contains the class implementation (all the functions)
- `appointment_main.cc`
  - Main program to test your class

Compiling your project manually:

1. `g++ -Wall -std=c++11 -c appointment.cc`
   - This creates the object file appointment.o
2. `g++ -Wall -std=c++11 -c appointment_main.cc`
   - This creates the object file appointment_main.o
3. `g++ appointment.o appointment_main.o`
   - This creates the executable `a.out`

or

```
g++ -Wall -std=c++11 -c appointment.cc appointment_main.cc
```

There is a Makefile provided to allow you to compile your program using the command:

```
make
```

You may use any function or library discussed in class or in the chapters we covered from your textbook. Do not use any other libraries or functions.

Hints:

- Write the default constructor, getters, and setters first. Then implement each of the helper functions one at a time.
- Test each function immediately after you write it.
- Write an output function to print all the values in an object.
- The function `stoi` allows you to convert a string object to an integer. It requires compiling your program with `-std=c++11` option.
  `int x = stoi("123");` will return the integer 123
- The function `to_string` converts a number to string.
  `string s = to_string(123);` will set s to the string `"123"`

- For the second constructor, use the `find` and the `substr` string member functions.

CS 2400 Appointment Class (60 Points) Due: Monday November 25, 2019
Part 1

Grading:
Programs that contain syntax errors will earn zero points.
Programs that do not include functions will also earn zero points.
Programs that use global variables other than constants will earn zero points.

You may use any function or library discussed in class or in the chapters we covered from your textbook. Do not use any other libraries or functions.

Your grade will be determine using the following criteria:

- Correctness (55 points)
  - (5 points) Default constructor
  - (15 points) Second constructor
  - (10 points) Getters and setters
  - (10 points) militaryToStandard function
  - (10 points) standardToMilitary function
  - (5 points) == function
- Style & Documentation (5 points)

Follow the coding style outline on GitHub:
https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md

Submit a link to your repository on Blackboard.