PIC 10B Lectures 1 and 2 Spring 2015
Homework Assignment #1

Due Wednesday, April 8, 2015 by 6:00pm.

Objectives:

1. To define classes using C++ inheritance.
2. To use C++ polymorphism to make your code more extensible.

Introduction:

So many appointments…so little time. Let's get organized and write a C++ application that simulates an interactive appointment book. To get started, we must model what an appointment is by defining an Appointment class and subclasses for various types of appointments: `OneTime` (for an appointment occurring only once), `Daily` (for an appointment that occurs every day), `Monthly` (for an appointment that occurs once a month on the same day), and `Yearly` (for an appointment that occurs once a year on the same month and day). The specifications for the assignment are below:

The class `Appointment` will have the following characteristics:
Attributes:

- `Date` object member called `date` (Date class code is provided.)
- `string` object member called `description`
- hour (of appointment)
- minute (of appointment)

Private helper member function `convertInt()` given below which is used to convert an int to a string: (Note: you must include the `stringstream` library `<sstream>` in your Appointment.cpp file)

```
string Appointment::convertInt(int number) const
{
   stringstream ss; //create a stringstream
   ss << number;    //add number to the stream
   return ss.str(); //return a string with the stream content
}
```

Constructor :

- `Appointment(string description, int month, int day, int yr, int hr, int min)` Use some of its parameters to construct the `date` member in its initializer list.

Behaviors:

- an accessor getDate() which cannot change the calling object nor its date object member, but just return it. Method getDate() should only be accessible to Appointment and its derived classes.
- a pure virtual function
  ```
  bool occurs_on(int month, int day, int year)
  ```
  that returns true if the appointment occurs on the given date (month, day, and year) and false otherwise.
- A const virtual function print() which displays the appointment in the console. For example, if description is "Gym" and the time of the appointment is10:30, print() should display
  ```
  10:30 -> Gym
  ```

The derived classes must all override both inherited virtual methods. Each derived class's version of print() will call its Appointment version and then output the type of appointment in parentheses (see screenshots below.) Overridden versions of function occurs_on can return true for month, day, and year arguments that fall prior to the Appointment's date for appointments that are regular (eg Daily, Monthly, or Yearly) provided the other criteria for such appointments are met.

Directions:

1. Create the project called "Hw1" in a solution called "Homework" using Microsoft Visual Studio 2012. All header (.h) and source (.cpp) files inside the project should contain the following header:

```
/*
   <Your Name>                    PIC 10B Intermediate Programming
   ID: <Your Student ID>          Spring 2015
   Email: <Your Email Address>    Assignment #1
   Section: <Your Section # eg 1A>
   Honesty Pledge:

      I, <Your Name Here>, pledge that this is my own independent work,
   which conforms to the guidelines of academic honesty as described in
   the course syllabus.

   List of known bugs:  <Known bugs, if any>
*/
```

2. Download the following files and add them to your project: Date.h, Date.cpp, and AppointmentBookApp.cpp. You will also be defining some functions in AppointmentBookApp.cpp to complete the app.

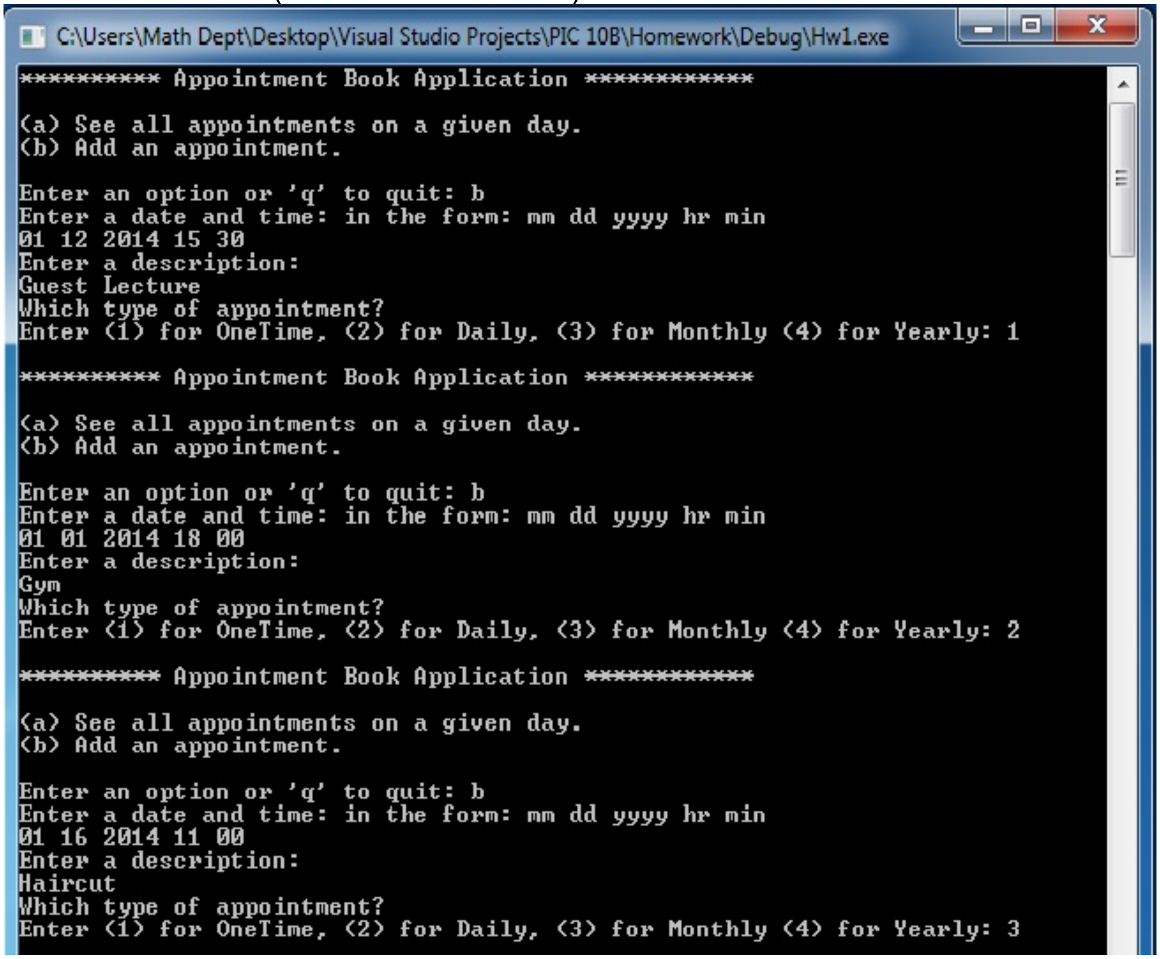
3. Define and implement base class `Appointment` and (publicly) derived classes `OneTime`, `Daily`, `Month`, and `Yearly` as described in the introduction. Create separate interface (.h) files and implementation (.cpp) files for each class. These files have the same name as the class as in the Polymorphism example on CCLE. The derived classes must all override both virtual inherited methods.

4. Complete application file AppointmentBookApp.cpp by implementing the following functions used in `main()`:
   - `void checkAppointments(const vector<Appointment*>& apptbook)` This function prompts the user for a date in the form mm dd yyy (eg 01 02 2014), reads that info from the keyboard (no input validation needed), loops through the given `apptbook`, and prints out only those appointments who occur on that day (using the recent most overridden print() function of the Appointment object pointee).
   - `void addAppointment(vector<Appointment*>& apptbook)` This function prompts the user for a date and time in the form mm dd yyy hr min (eg 01 02 2014 16 45), reads that info from the keyboard, prompts and stores the appointment description, and dynamically creates (on the heap) a user-specified subtype of Appointment and adds a pointer to it in the vector `apptbook`.

5. When you have completed your project, be sure to
   - make sure your program compiles in Visual Studio 2012.
   - run your program to make sure it works correctly
   - upload your source code files
     - Appointment.h,
     - Appointment.cpp,
     - OneTime.h
     - OneTime.cpp
     - Daily.h,
     - Daily.cpp,
     - Monthly.h,
     - Monthly.cpp,
     - Yearly.h,
     - Yearly.cpp, and
     - AppointmentBookApp.cpp

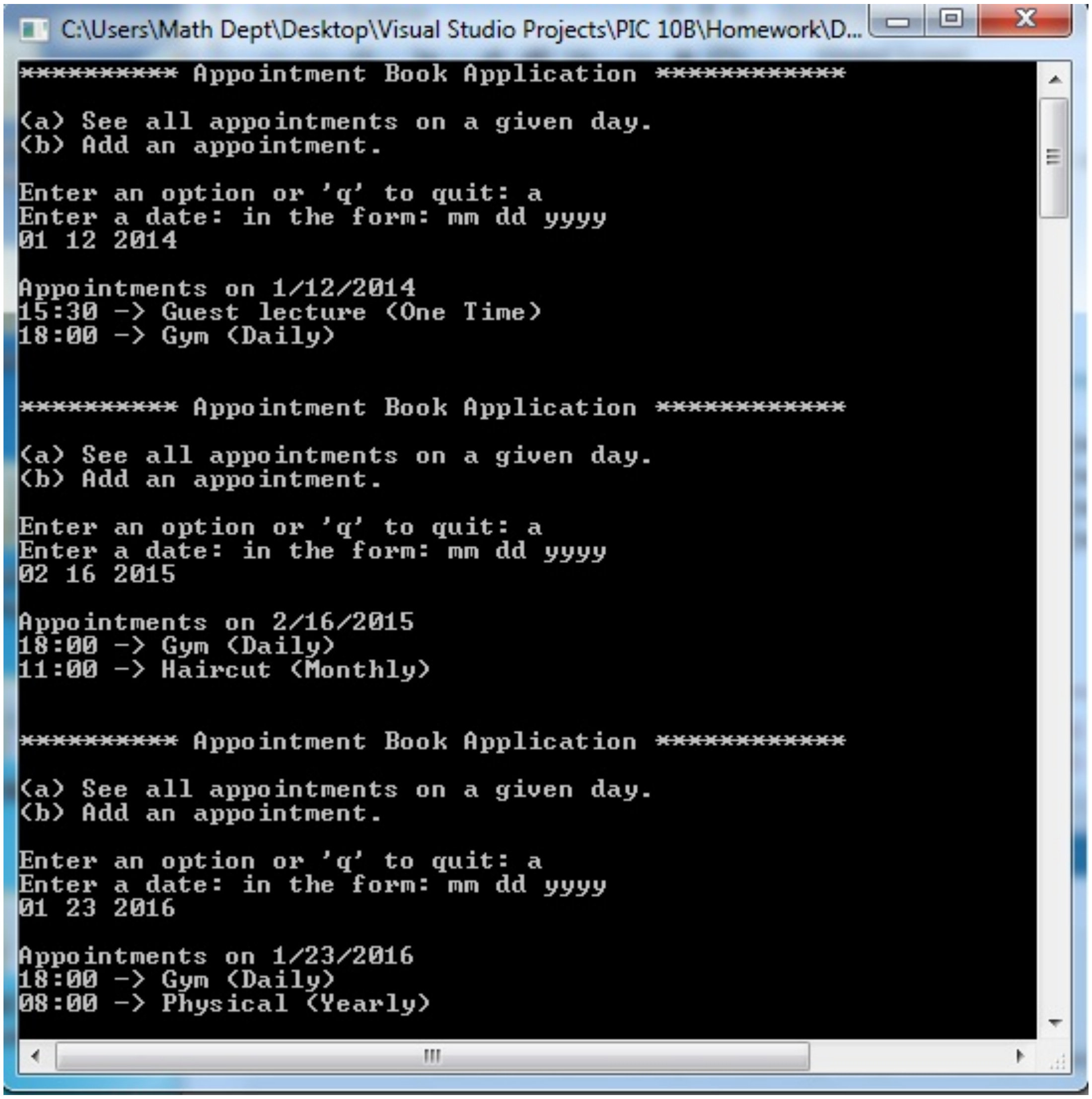
     using the CCLE website. No hardcopies will be collected.
   - visually verify that your source code was submitted correctly by clicking on the links to those files on the CCLE page after submission.

Some screenshots (all the same session) are below:



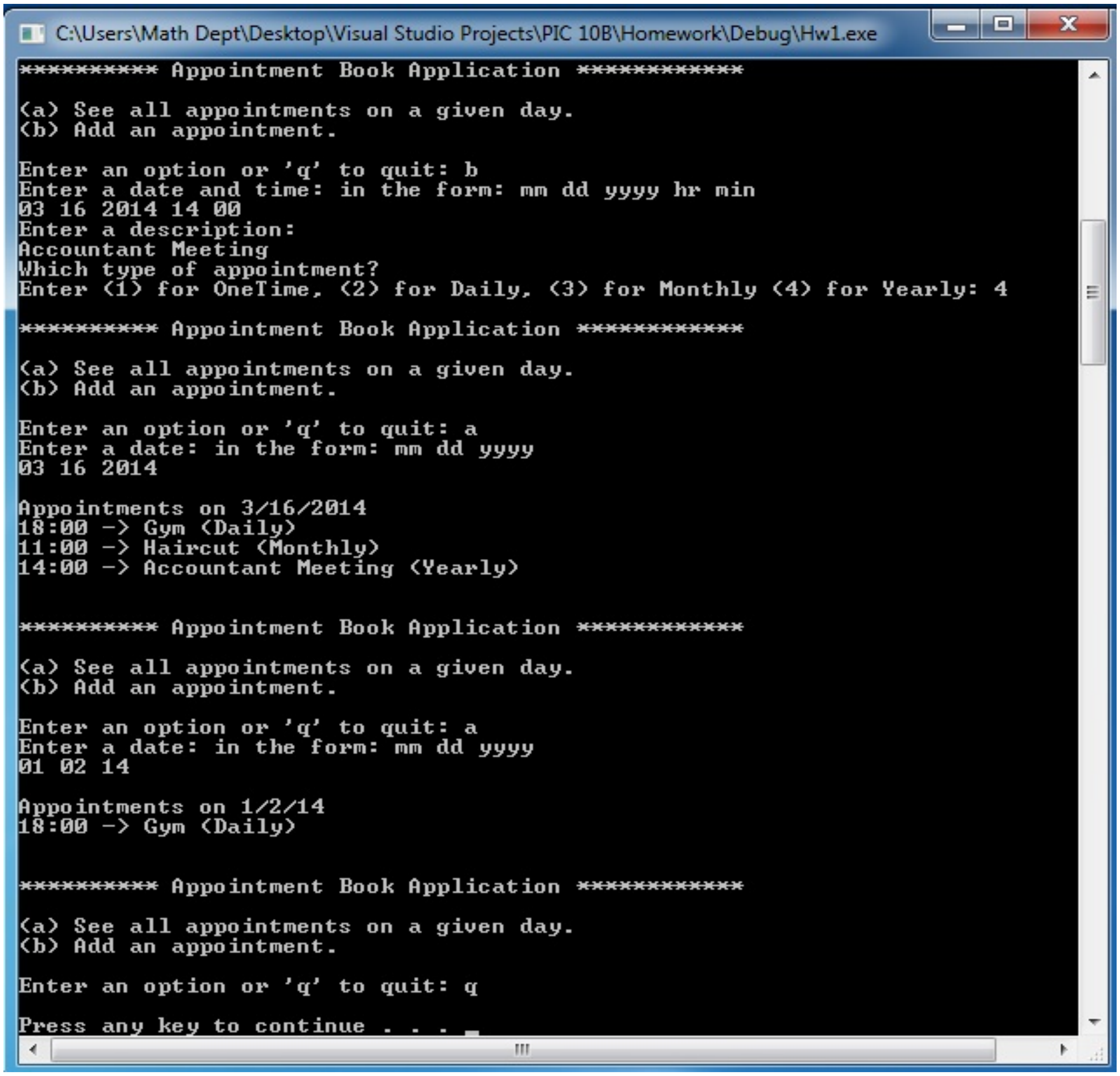
C:\Users\Math Dept\Desktop\Visual Studio Projects\PIC 10B\Homework\Debug\Hw1.exe

```
********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: b
Enter a date and time: in the form: mm dd yyyy hr min
01 12 2014 15 30
Enter a description:
Guest Lecture
Which type of appointment?
Enter (1) for OneTime, (2) for Daily, (3) for Monthly (4) for Yearly: 1

********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: b
Enter a date and time: in the form: mm dd yyyy hr min
01 01 2014 18 00
Enter a description:
Gym
Which type of appointment?
Enter (1) for OneTime, (2) for Daily, (3) for Monthly (4) for Yearly: 2

********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: b
Enter a date and time: in the form: mm dd yyyy hr min
01 16 2014 11 00
Enter a description:
Haircut
Which type of appointment?
Enter (1) for OneTime, (2) for Daily, (3) for Monthly (4) for Yearly: 3
```



C:\Users\Math Dept\Desktop\Visual Studio Projects\PIC 10B\Homework\D...

```
********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: a
Enter a date: in the form: mm dd yyyy
01 12 2014

Appointments on 1/12/2014
15:30 -> Guest lecture (One Time)
18:00 -> Gym (Daily)


********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: a
Enter a date: in the form: mm dd yyyy
02 16 2015

Appointments on 2/16/2015
18:00 -> Gym (Daily)
11:00 -> Haircut (Monthly)


********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: a
Enter a date: in the form: mm dd yyyy
01 23 2016

Appointments on 1/23/2016
18:00 -> Gym (Daily)
08:00 -> Physical (Yearly)
```


C:\Users\Math Dept\Desktop\Visual Studio Projects\PIC 10B\Homework\Debug\Hw1.exe

```
********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: b
Enter a date and time: in the form: mm dd yyyy hr min
03 16 2014 14 00
Enter a description:
Accountant Meeting
Which type of appointment?
Enter (1) for OneTime, (2) for Daily, (3) for Monthly (4) for Yearly: 4

********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: a
Enter a date: in the form: mm dd yyyy
03 16 2014

Appointments on 3/16/2014
18:00 -> Gym (Daily)
11:00 -> Haircut (Monthly)
14:00 -> Accountant Meeting (Yearly)


********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: a
Enter a date: in the form: mm dd yyyy
01 02 14

Appointments on 1/2/14
18:00 -> Gym (Daily)


********** Appointment Book Application ************

(a) See all appointments on a given day.
(b) Add an appointment.

Enter an option or 'q' to quit: q

Press any key to continue . . . _
```

Grade Breakdown:

Criteria	Description	Points
Header	Starts every .h and .cpp file.	1
Comments	Program well-commented.	1
Appointment	Class defined and implemented as specified.	6
OneTime, Daily, Monthly, Yearly	Classes derived publicly from Appointment. Classes correctly override both virtual inherited methods occurs_on and print.	12
AppointmentBookApp.cpp	`checkAppointment` correctly defined. Uses dynamic method binding.	5
AppointmentBookApp.cpp	addAppontment correctly defined.	5
Total		30

A penalty of 5 points will be assessed if your code does not compile using Microsoft Visual Studio 2012.