

# Assignment 5

## Simple Appointment Book

### Due Friday, May 2, 2014, 11:59pm

#### Introduction:

This is a relatively easy assignment to help you practice polymorphism and virtual functions. You can think of this as the calm before the storm of the final project. Since this is a pretty easy assignment, it **MUST** be done individually.

Your job is to implement 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.

The class `Appointment` will have the following characteristics:

#### Attributes:

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

#### Constructor :

- `Appointment(string description, int day, int month, 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 day, int month, 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 "OOP Class" and the time of the appointment is 10:30, `print()` should display

10:30 -> OOP Class

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 example interaction at the end of this handout).

#### Directions:

1. Download the following files: `Date.h`, `Date.cpp`, and `AppointmentBookApp.cpp`. You will also be defining some functions in `AppointmentBookApp.cpp` to complete the assignment.

2. Define and implement base class `Appointment` and publicly derived classes `OneTime`, `Daily`, `Month`, and `Yearly` as described above. Create separate interface (.h) files and implementation (.cpp) files for each class. The derived classes must all override both virtual inherited methods.

3. 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 dd mm yyy (e.g. 01 02 2014), reads that info from the keyboard, loops through the given apptbook, and prints out only those appointments which occur on that day (using the overridden print() function of the Appointment object pointeer).

- void addAppointment(vector<Appointment\*>& apptbook)

This function prompts the user for a date and time in the form dd mm yyyy hr min (eg 01 02 2014 16 45), reads that info from the keyboard, prompts and stores the appointment description, prompts and stores the appointment type, and dynamically creates (on the heap) a subclass of Appointment and adds a pointer to it in the vector apptbook.

4. When you have completed your project, be sure to

- compile and run your program to make sure it works correctly in Linux.

- zip your source code files

Appointment.h

Appointment.cpp

OneTime.h

OneTime.cpp

Daily.h

Daily.cpp

Monthly.h

Monthly.cpp

Yearly.h

Yearly.cpp

AppointmentBookApp.cpp

and upload them to LMS.

#### **Grade Breakdown:**

Criteria	Description	Points
Comments	Program well-commented.	5
Appointment	Class defined and implemented as specified	15
OneTime, Daily, Monthly, Yearly	Classes derived correctly from Appointment. Classes correctly override both virtual inherited methods occurs_on() and print().	40
AppointmentBookApp.cpp	checkAppointment correctly defined. Uses dynamic method binding.	20
AppointmentBookApp.cpp	addAppointment correctly defined.	20
Total		100

#### **Acknowledgements:**

This assignment is a variation of an assignment designed by Ian Charlesworth at UCLA for an intermediate programming course.

### **Example Interaction:**

```
student@student-VirtualBox:~/cs212/asgn5$ ./apptbook
***** Appointment Book Application *****
```

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

```
Enter an option or 'q' to quit: b
Enter date and time (dd mm yyyy hr min): 12 1 2014 15 30
Enter description:
Guest Lecture
Enter appointment type: (1) for one-time, (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 date and time (dd mm yyyy hr min): 01 01 2014 18 00
Enter description:
Jog
Enter appointment type: (1) for one-time, (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 date and time (dd mm yyyy hr min): 16 1 2014 11 00
Enter description:
Haircut
Enter appointment type: (1) for one-time, (2) for daily, (3) for monthly, (4)
for yearly: 3
```

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

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

```
Enter an option or 'q' to quit: a
Enter date (dd mm yyyy): 12 1 2014
Appointments on 12/1/2014:
15:30 -> Guest Lecture (One Time)
18:00 -> Jog (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 date (dd mm yyyy): 16 2 2015  
Appointments on 16/2/2015:  
18:00 -> Jog (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 date (dd mm yyyy): 23 1 2016  
Appointments on 23/1/2016:  
18:00 -> Jog (Daily)

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

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

Enter an option or 'q' to quit: b  
Enter date and time (dd mm yyyy hr min): 23 1 2014 08 00  
Enter description:  
Physical  
Enter appointment type: (1) for one-time, (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 date (dd mm yyyy): 23 1 2016  
Appointments on 23/1/2016:  
18:00 -> Jog (Daily)  
8:00 -> Physical (Yearly)

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

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

Enter an option or 'q' to quit: b  
Enter date and time (dd mm yyyy hr min): 16 3 2014 14 00  
Enter description:  
Accountant Meeting  
Enter appointment type: (1) for one-time, (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 date (dd mm yyyy): 16 3 2014  
Appointments on 16/3/2014:  
18:00 -> Jog (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 date (dd mm yyyy): 2 1 2014  
Appointments on 2/1/2014:  
18:00 -> Jog (Daily)

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

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

Enter an option or 'q' to quit: q