# COSC 190 (2017)

# Assignment 1

# Due Date Jan 25th

Submission Requirements

All of the required classes will be contained in a Project called Assign1Cst1XX where 1XX corresponds to your CST Number. In addition all classes will further be members of a package called *assign1*. You will hand in your assignments to the Directory A:\CST\COSC190\Assignment1. You are not required to print out your assignment.

# The Appointment Class [25 Marks]

You are required to implement an Appointment class and sub classes as per the UML diagram Presented on Page 2 of this assignment (also saved as Appointment.dia). This class is to represent appointments that an individual might make for their Calendars. One Time appointments are just that – single appointments on the calendar. Daily appointments occur every day at the same time, while Monthly appointments occur once per month.

The method occursOn is actually an abstract method and must be implemented for each of the subclasses.

You can assume the following for the Appointment classes

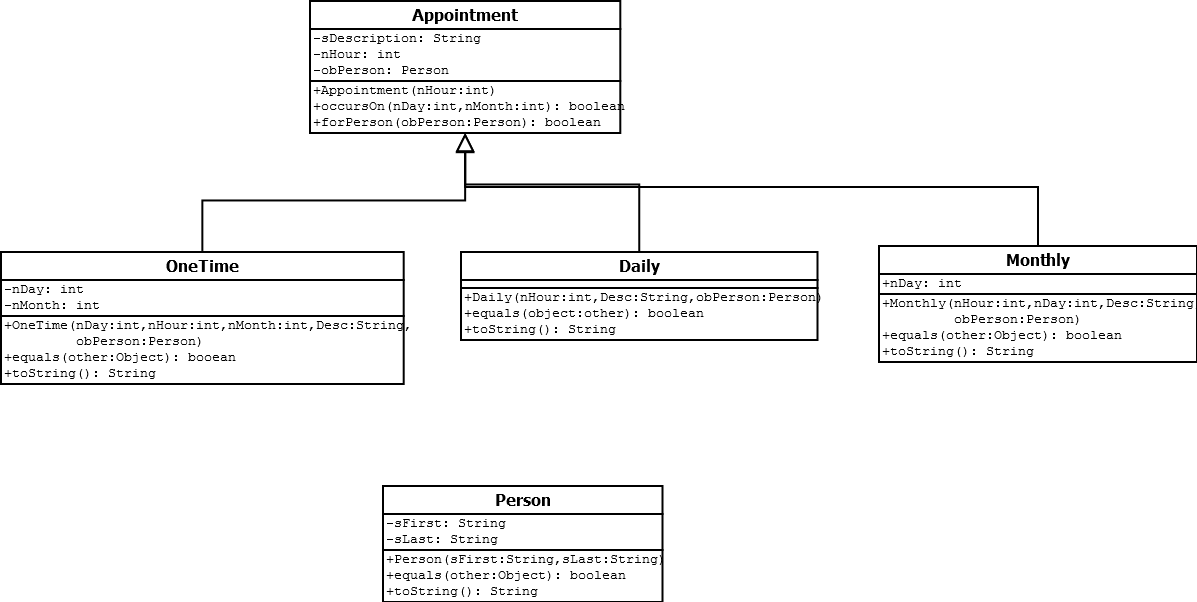
* All Appointments are 1 hour long
* Appointments can be booked from 8:00 to 17:00.
* The Month Attributes will refer to the Month with 1 being “Jan”, 2 being “Feb” and so on.

Notes:

equals - For all Appointment classes, the equals method is checking against the Time/Hour/Day of the appointment – recognize that for Daily – matches all days. For the Person class, equals only compares the sFirst and sLast Names against each other.

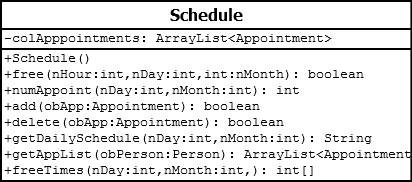
occursOn – As indicated is an abstract method that must be handled in the sub-classes.

forPerson – This method, given a Person Object returns true if the indicated appointment is for the given person.



# The Schedule Class [50 Marks]

You are further required to create a Schedule class that will be responsible for managing appointments. The UML Diagram for this is given below:



Notes for Schedule

* ***free*** – This method returns true if there is no appointment in the list that corresponds to that time
* ***numAppoint*** - This method returns the number of appointments that occur on the indicated day
* ***add*** – This method attempts to add an appointment to the schedule. If an appointment already exists in the list for specified time then do not add the appointment and return false, otherwise return true.
* ***delete*** – This method attempts to delete an existing appointment at the specified time. It returns false if no ***appointment*** exists at the given time
* ***getAppList –*** This returns an ArrayList of Appointments for a particular Person- should be every appointment that person has stored in the schedule.
* ***freeTimes*** – This returns an array of all the open times (hours) on the given day.
* ***getDailySchedule -***  This will return a String that when printed will produce a “nice” output of the schedule for that Day. For Example consider the following partial schedule as an acceptable output form:

***8:00***

***9:00 Broken Leg – Rob Miller***

***10:00 Check Up - Sharon McDonald***

***11:00 Lunch -***

***12:00 Family Counseling - Ron New***

***……………………………………….***

# An Interface [15 Marks]

Have the Appointment class implement the following interface

*public interface EarlyMorning()*

*{*

*boolean getEarly();*

*}*

The method getEarly should return true if the given Appointment is actually an early appointment (**before** 10:00 AM).

Using this modify the schedule class to have a new method whose signature is:

Appointment[] getEMorning(int nDay,int nMonth) which return an array of appointments for the given Day/Month.

# Testing: [10 Marks]

You should write a program that tests all of your methods. Book at least 40 Appointments over a few day span and check to see that the routines are working correctly.

Final Notes:

Feel free to add any methods to the above classes that you feel that they need. I will be testing your program by dropping the sources files into my own package. Use the same Class/Method names as provided in this assignment – so follow the UML diagrams exactly.