

# INFSCI 0020 Program Design and Software Tools

## Homework 5

Due July 12

Total points: 200

### *Part 1*

This assignment involves

- Using the code from the book and making some modifications and/or additions.
- Using a very useful Standard Library Class **Vector**

Each part will constitute 50 points.

1. Book's problem 9.3 (page 660)

- a. Here you do not actually have to rewrite class `Point3`, but you need to use this class for others.
- a. Your program should produce the same output as the original one.

Book's example code has been provided for download. There are two zip files.

- b. File `HW4-1a.zip` contains code to test `Circle4`
- c. File `HW4-1b.zip` contains code to test `Cylinder`

Note that in code for `Cylinder`, the class for `Point3` and `Circle4` classes from `HW4-1a.zip` are used.

**Submission:** You will be submitting two zip files as above (as `HW4-1a.zip` and `HW4-1b.zip`) – the file names and number of files will be the same as in the provided zip files. You will only have changed the code of these files.

2. Book's problem 10.12 (page 717)

- a. For this assignment you will need to re-read pages 592-596 (re-reading Chapter 8 is highly recommended)

Book's example code has been provided for download. There are two zip files.

- d. File `HW4-2a.zip` contains code for the example in Figure 10.23-10.33
- e. File `HW4-2b.zip` contains code to for example on `Date` and `Employee` from chapter 7.

You will use the `Date` class provided in `HW4-2b.zip` file and modify files in `HW4-2a.zip` file.

**Submission:** You will submit `HW4-2a.zip` file. This will include files related to `Date` class and files of `HW4-2a.zip` with needed modification.

## ***Part 2: Deque, Stack and Queue***

### **Deque Implementation**

Files *list.h* and *listnode.h* have been provided for this exercise. By using these files, you will first implement the **Deque** class. You may want to read carefully the **Stack** implementation in Chapter 17 using *list.h* and *linknode.h* – it will be a simple modification of the **Stack** implementation.

A **Deque** object allows insertions and deletions at both the *front* as well as the *back* of a list and hence is sometimes called a *double ended queue*. The key member functions needed for a **Deque** class include;

<b>pushFront</b>	: insert an element in the front of the <b>Deque</b>
<b>pushBack</b>	: insert an element in the back of the <b>Deque</b>
<b>popFront</b>	: removes and returns an element from the front of the <b>Deque</b>
<b>popBack</b>	: removes and returns an element from the back of the <b>Deque</b>

Additionally, include the following support functions:

<b>getSize</b>	: return the size of the <b>Deque</b>
<b>isDequeEmpty</b>	: returns true if the <b>Deque</b> is empty
<b>printDeque</b>	: prints the elements of the <b>Deque</b> from front

You can define any additional member function as you deem fit. You need to define appropriate member variables – a look at the **Queue** class in Chapter 17 of the book will help you decide appropriate member variables.

### ***Client program for Deque :***

Write a client program that will allow testing for all the functions. Provide the following *menu* to allow choosing the operations users want to perform:

1. Create a deque for user defined type (e.g., float, double)
2. Push an element at the front of the **Deque**
3. Push an element at the back of the **Deque**
4. Pop an element from the front of the **Deque**
5. Pop an element from the back of the **Deque**
6. Print the size of the **Deque**
7. Print the elements of the **Deque**
8. Exit the program

Your client program should demonstrate these functionalities for **int** and **double**.

### ***Stacks and Queues implementation***

Now on top of the **Deque** class, build the **Stack** and **Queue** classes. For **Stack** implementation refer to Chapter 17. Only difference now is that, **Stack** class does not need to know about *list.h* and *listnode.h* files. **Queue** class should allow only adding in the front and retrieving elements from the back. Read Fig. 21.24 for a queue client.

Your client programs for the **Stack** and **Queue** implementation should demonstrate the functionalities for **int** and **double**.