**ASSIGNMENT SET 6**

**Date: Thursday, 5/03/2020**

**Submission Deadline: Sunday 15/03/2020**

**Objective(s): Learn to implement the following features…..**

* **Static Members (data/methods), Static/Constant Objects, Constant methods and Constant and Reference data Members (use of Constructor Initializer list)**
* **Cascaded calls (use of this pointer)**
* **Template Functions/Class,**
* **Composition (has-a) relation among classes [Ex: hash class uses linked\_list class which in turn uses the node class]**

**In this assignment set instead of writing afresh, mostly (Assign 2 to 8) you have to convert your previous class definitions to class templates; minimally changing the int/float type to a generic symbol type.**

**==================================================================**

**1(a)** Implement a Salary class with components like basic, Da, bonus, from which you calculate the gross salary (private member) of an employee using the formula explained in the class lecture. Declare suitable members and objects to implement the static/constant/reference members as discussed.

**1(b)** Write a friend function to show how the average gross salary of two salary objects can be calculated using friend function.

**2.** Convert the node (single\_link) class you implemented in previous assignments to a suitabletemplate **node class** (single\_link), you may consider passing both symbol as well as non-symbol type parameters (as required).

Assign default arguments to template parameters.

3. Convert the linked\_list class you implemented in previous assignments to a suitable template **linked\_list class** using the template node (single\_link) class in Assignment 2.

4. Convert the stack class you implemented in previous assignments to a suitable template **stack class**, consider passing both symbol as well as non-symbol type parameters (as required). Assign default arguments to template parameters.

4. Convert the queue class you implemented in previous assignments to a suitable template **queue class**, consider passing both symbol as well as non-symbol type parameters (as required). Assign default arguments to template parameters.

5. Write a template **node** (double link) **class**, you may consider passing both symbol as well as non-symbol type parameters (as required). Assign default arguments to template parameters.

6. Convert the binary\_tree class you implemented in previous assignments to a suitable template **binary\_tree class**, using the node (double link) class; consider passing both symbol as well as non-symbol type parameters (as required). Assign default arguments to template parameters.

7. Convert the heap class you implemented in previous assignments to a suitable template **heap class**.

8. Convert the hash class to a template **hash class** which uses the template linked\_list class in Assignment 3.

**Bonus:**

**Use the template hash class in Assignment 8 to hash the roll sheet file as provided in the mail. Try using Universal Hash Function. One example of which :** if we are hashing strings of length k, then xi could be the ith character (assuming our table size is at least 256) or the ith pair of characters (assuming our table size is at least 65536). Furthermore, we will require our table size M to be a prime number. To select a hash function h we choose k random numbers r1,r2,... ,rk from {0, 1,... ,M − 1} and define: h(x) = r1x1 + r2x2 + ... + rkxk mod M. **Report the evenness of the load across the buckets by measuring the Mean & STD of the load.**