Lab Assignments – XI

MCA Semester III
CG and Java Lab (CS3307)

1. Create a generic class Calculator<T> to perform arithmetic of two same data type, i.e., Calculator<Integer> should perform arithmetic of two integers, Calculator<Double> should perform the same for doubles. Perform +, -, * an / for all. Throw an exception in modulus if the passed data type is not an integer, otherwise for Calculator<Integer> it must return the modulus.

2. Create a generic Stack class, where for Stack<T> the push operation will be "void push(T val) {}" and for pop operation, it is "T pop() {}". The constructor must take an integer *n* as input, that represents the size of the internal array. The push and pop operation must throw suitable exceptions on overflow and underflow [Note: Array of generic type cannot be made, i.e. T arr[] = new T[n] won't work, because Java needs to know the data type of the array before allocating storage for it. So, use an Object array internally. On push, simply push it and on pop, typecast it to T before returning].

3. Create a class named Person storing any information (such as name, address, job, office etc.). Create a List of several Persons. [Use List<Person> persons = new ArrayList<>(), and then add each person one by one]. Then traverse through the list and print the information of each person in the list.

4. Create a List of Strings in Java. Then using Set of Character, determine the number of unique characters in each String in the List.

5. Take a graph in a suitable input [Hint: take as adjacency list via an array of Lists as follows, where adjacency[i] contains adjacency list of i^th vertex]. Then use Queue Collection to perform BFS from any node, to obtain minimum distances to all other nodes.

```
ArrayList<Integer>[] adjacency = new ArrayList[n];
for(List<Integer> vertex : adjacency) {
    vertex = new ArrayList<Integer>();
}
```