Lab Assignments – III

MCA Semester III
CG and Java Lab (CS3307)

1. Create an interface named *DBEntity* inside a package *inst.nitjsr.hospitals.dblayer*. The interface contains abstract methods such as *insert, delete, update*. The *insert* operation takes a *DBEntity* type variable as parameter, *delete* operation takes an integer as a parameter, and *update* operation takes two parameters, an integer and a *DBEntity* variable. Now, implement this interface into three concrete classes *Doctor, Staff, Patient*. In each implementation, for now, just show a message such as "from insert method of Doctor class" [You can guess the possible strings for other method implementations in other concrete classes]

2. Now add some properties to each of these classes, such as in Doctor class, you may add *id, name, specialization, chamber_floor* etc. (You may similarly place properties in other concrete classes as it seems best suited from your viewpoint. Each class must have atleast an id property). Create setter methods for these properties in the classes. Now in the implemented interface methods, typecast the *DBEntity* parameter into that corresponding concrete class, and change the message accordingly. As for example, the *insert* will print "Inserting doctor with id: <> into database", *delete* will print "Deleting doctor with id: <> from database", *update* will print "Modifying information of doctor with id: <> in database".

3. Create an appropriate static factory method in DBEntity, that will return one of the concrete classes. From main method, employ these concrete classes accordingly. Such as the following sequence should print "Inserting doctor with id 1 into database".

```
Doctor d = (Doctor)DBEntity.getEntity("doctor");
d.setId("1");
d.setName("A");
d.setSpec("Neuro");
d.insert(d);
```

4. Declare a new method named *alter* in *DBEntity* interface in such a way that this method does not need to be implemented in the concrete classes. Invoke this method from the concrete subclass objects in main, they should print "Implementation in progress".

5. Now perform the same problems from 1-4 by converting the *DBEntity* from interface to an abstract class.