Lab Assignments – II

MCA Semester III
CG and Java Lab (CS3307)

1. Create a Vector class with overloaded constructor. It must have atleast two constructors, one taking an array as input, that will set the elements of the vector (internally, the class will maintain an array of the elements). The other constructor will take another Vector as input, and will create a copy of that. The class must have a "void set (int index, double value)" function setting the element at index "index" to value "value" and a "double get(int index)" function returning the element at index "index".

2. Implement two overloaded "multiply" methods in Vector class, one accepting a double element performing multiplication of a vector with a scalar, and the other one accepting another Vector performing dot product.

3. Implement a class named Employee with four properties *name, id, office* and *designation.* This class must have a single constructor setting these four properties (use *this* keyword). This class has two subclasses *Developer* and *Manager.* They must also have their own constructors taking only name, id and office. They will set the designation to "developer" and "manager" respectively.

4. Modify the Employee class to be an abstract class by adding two abstract methods *insert()* and *delete().* The insert method in Developer will just print "Data inserted in Developer database" and delete method will print "Data deleted from Developer database". Similar implementation will be performed in Manager class as well. (The intended implementation will be performed after discussion of JDBC in theory class). In Employee class, create a factory method named getEmployee(String type) that will return either a Developer or a Manager object, depending on the value of type. In main method, perform Dynamic Method Dispatch by assigning different subclss objects in same Employee variable using the factory method, and invoking the abstract methods from the different subclass objects accordingly.

5. Create a static variable *count* and static method *getEmployeeCount*() that will return the number of Employee objects created thus far (Hint: increase the static variable each time an Employee object is created).