<u>Lab Assignments – V</u>
MCA Semester III
CG and Java Lab (CS3307)

1. Create an interface named *Iface1*. This interface must have at least one undefined method *im1()*. Create one private method of your choice, and create a default method named *dm1()* in the interface. Now, let a concrete class named *Concrete* to implement the interface. Create an object of the concrete class, and invoke *im1* and *dm1* from it. *Concrete* must implement *im1*, but it should not override *dm1*.

2. Now create another concrete class named *Concrete2* that implements the same interface *Iface1*. Implement *im1* and override *dm1* in it. Now create a static factory method inside *Iface1*, and employ it to dynamically create objects of *Concrete* and *Concrete2*. Invoke *im1*, and *dm1* from the objects accordingly.

3. Create another interface named *Iface2*. Create two undefined methods *im1()* and *im2()*, and a default method named *dm1()*. Now let both the concrete classes implement both the interfaces. Only implement *im2()* in the two concrete classes. Compiling the java files should now give error in one concrete class, but not in another. Why? Correct the error-producing class accordingly.

-------------------- Problems 4 and 5 are independent from first 3 --------------------------------

4. In main method, invoke another method of your choice, that will perform some division by zero. Do not catch the exception. Note down the stack trace printed by JRE. Now, catch the corresponding exception, and print the exception string from within the catch clause. Program must not terminate in this second scenario.

5. Surround the try block with another catch clause of your choice, say ArrayIndexOutOfBoundException. Access some out of bound array index in your catch clause. Swap the sequence of previous method call and array access, and observe which catch clause gets invoked.