**Excersize 1: Cover the first part of headway program.**

**Problem Statement 1**: cover JDBC and programming fundamentals.

1. Write a java program that connects to database Postgres SQL with database name “Company”.
2. Create table named “Student” in the created database.
3. Create table named “Departments” that indicates the departments of these student.
4. Student table fields: Id, name, age, grade, department, faculty\_serial\_number.
5. Department table fields: name, id, Boss
6. The relation is many-to-many. One student is in many departments, and 1 department includes many students.
7. The join table name is : students\_departments
8. Add unique constraint on student\_facuilty\_serial number.
9. The program will do the following:
   1. Select all students.
   2. Select all students in specific department by inner join.
   3. Update student info by Id.
   4. Delete student by id.
   5. Get count of students in departments
   6. List all students that has no departments.
10. Allow the program to through unique constraint violation exception.

**Problem statement 2:** write a recursive function that prints the following sequence n n\*2 n\*3 n\*4 n\*5 …..n\*10 until reach 10\*n then prints the backward sequence until n like : n\*10 n\*9 ……n don’t use loops

**Problem statement 3:** Suppose that you start with 250 units. Then you could make these moves:

--Start with 250 units.

--Since 250 is divisible by 5, you may return 42 of the units, leaving you with 208 units.

--Since 208 is even, you may return half of the units, leaving you with 104 units.

--Since 104 is even, you may return half of the units, leaving you with 52 units.

--Since 52 is divisible by 4, you may multiply the last two digits (resulting in 10) and return these 10 units. This leaves you with 42 units.

--You have reached the goal!

Write a recursive method to meet this specification.

**Problem statement 4**: create a java program that creates class “Rectangle” and compare different instances (polymorphism) of this class using comparable interface.

**Problem statement 5:** Write a Java program to create a class called "Bank" with a collection of accounts and methods to add and remove accounts, and to deposit and withdraw money. Also define a class called "Account" to maintain account details of a particular customer. Create a class “Customer” and linked it with his accounts.