

**Ahmedabad University**  
**School of Engineering and Applied Science**  
**B. Tech. (ICT) Semester – II**  
**CSC103 – Object Oriented Programming Lab**  
**Lab Assignment – 4**

---

Last Date of Submission: 20<sup>th</sup> March 2019

---

**Topics covered:** Java Database Connectivity, SQL, Class diagram

**Important Instructions:**

Follow below coding and naming conventions while developing your programs:

1. Class names should be nouns and first letter must be capital letter. The first letter of each internal word capitalized. Use whole words-avoid acronyms and abbreviations E.g. BankAccount, Employee, AdminStaff, SavingAccount
2. Methods should be verbs with camel case, first letter lowercase and the first letter of each internal word capitalized.  
E.g. addEmployee( ); deleteEmployee( ); displayEmployeeList( );
3. Variables: Variable names should be meaningful. E.g. for Employee Class, variable names should be employeeId, employeeName, employeeDesignation, employeeScaleOfPay, etc.
4. Every Program/SQL query document should have header having following information  
/\*@author RollNo. - Firstname Lastname @version dd/mm/yyyy  
Description: Write program/query/class diagram definition.  
\*/  
Every Program/SQL query should have footer with output of the query (if applicable)  
/\* PROGRAM OUTPUT \*/

**Policy on Plagiarism (Copying of code/programs) and academic dishonesty**

Copying Lab assignment/program code doesn't help you to learn the concepts.

Programs designed, developed and submitted by a student should be the result of original and individual work based on his/her own efforts. Full or part of the code should not be copied from internet or from peer students or other sources. A student should not share/circulate the code/programs developed by them (for individual assignments) with their peers in any form. Violation of above will be considered as academic dishonesty and any such case will be strictly dealt with and liable to get zero in the evaluation.

## Exercises based on Structured Query Language (SQL)

---

This laboratory exercise helps you to understand the *basic concepts* of Structured Query Language (SQL). Understanding of basic SQL statements is important to develop JDBC related applications.

Note: Describe table structure as shown below along with SQL queries:

tbl\_customer

Column	Type	Null	Default	Comments(if any)
customer_id (Primary)	int(11)	No		
name	varchar(30)	No		
city	varchar(30)	No		

### Exercise #1 Based on Customer table.

Create a database (name: MyDatabase1) in MySQL software available under WAMP software. Create a **Customer** table to store following information about the customers.

CustomerID	CustomerName	ContactNo	Address	City	PostalCode	Country
------------	--------------	-----------	---------	------	------------	---------

- Take suitable data types while creating tables.
- Insert 5 customer records into Customer Table using GUI facility of MySQL.

### Write SQL statement/query for the followings:

- Insert 5 customer records into Customer Table using INSERT statement.
- Display information of all customers from Customer Table using GUI facility of MySQL and using SELECT statement.
- Display information of all Customers from city Ahmedabad.
- Display Customer name and Contact number of all Customers from India.
- Display total number of customers.
- Display customer information whose name begins with "A".

### Exercise #2 Based on Employee table.

Create an **Employee** table based on information/query given below in MySQL:

```
CREATE TABLE employees  
( employee_number int NOT NULL,
```

```
last_name char(50) NOT NULL,  
first_name char(50) NOT NULL,  
salary int,  
dept_id int,  
CONSTRAINT employees_pk PRIMARY KEY (employee_number) );
```

- Add 10 suitable records using INSERT query. One such example is given below:

```
INSERT INTO employees (employee_number, last_name, first_name, salary, dept_id)  
VALUES (101, 'Patel', 'K', 650000, 10);
```

- a. Display information of all employees.
- b. Display information of all employees with salary > Rs. 10,00,000/-.
- c. Display information of all employees with salary in range of Rs. 5,00,000/- to Rs. 10,00,000/-
- d. Display information of all employees working in department no. 10.
- e. List the employees who does not belong to department no. 10.
- f. Display total number of employees working in the department no. 10.

### **Exercise #3 Based on Supplier table.**

Create a table to store information about Suppliers as shown below:

```
CREATE TABLE suppliers  
( supplier_id int NOT NULL,  
supplier_name char(50) NOT NULL,  
city char(50),  
state char(25),  
CONSTRAINT suppliers_pk PRIMARY KEY (supplier_id) );
```

- Add 10 suitable records using INSERT query as shown below:

```
INSERT INTO suppliers (supplier_id, supplier_name, city, state)  
VALUES (101, 'Microsoft', 'Redmond', 'Washington');
```

- a. Display information about all supplier from city Redmond.
- b. List the information of all supplier who are from the city “Redmond” or “New York”.
- c. Display information about all supplier in descending order by supplier name.
- d. Select the unique city values that reside in the state of Washington and display them in descending order by city. [Hint: use DISTINCT keyword]

#### Exercise #4 Based on Customer and Order tables (Use of multiple tables from database).

Create an **Order** table to store information as shown below:

OrderID	CustomerID	OrderDate
103	1	2019-01-18
104	2	2019-01-19
105	3	2019-01-20

Create a **Customer** table to store information as shown below:

CustomerID	CustomerName	ContactNo	Country
1	Amit Patel	9898989898	India
2	Chin Chen	9797979797	China
3	Sunit Shah	9897989799	India

Write the SQL query to display information as shown below:

OrderID	CustomerName	OrderDate
103	Amit Patel	2019-01-18
104	Chin Chen	2019-01-19
105	Sunit Shah	2019-01-20

Notice that the "CustomerID" column in the "Orders" table refers to the "CustomerID" in the "Customers" table. The relationship between the two tables above is the "CustomerID" column.

Hint: A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

\*\*\*\*\*

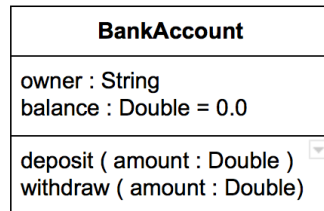
## Exercises based on Class diagrams

---

### Notes: Class Representation in UML

- A class is represented as a box with 3 compartments.
- The uppermost one contains the class name.
- The middle one contains the class attributes and the last one contains the class methods.

### Example of Class diagram:



Consider the following Scenarios. Identify entities with attributes and relationships between entities. Draw class diagram for each scenario.

Prepare your class diagram using any convenient tools as discussed in OOP theory class.

Submit a word document containing image of your class diagram.

---

### Exercise #5 Banking System

Draw a class diagram (for Banking System) that models the relationships between the following classes: Bank, Savings Account, Current Account, Loan, ATM, Customer (Feel free to add more classes if needed while preparing Class diagram).

### Exercise #6 University Management System

A university contains many schools. Each school can offer any number of courses. Each school employs many instructors, but an instructor can work in only one school. An instructor can have many teaching assistants, but a teaching assistant can assist only one instructor. For each school, there is a dean. An instructor can be dean of any school. Each instructor can teach any number of courses. Each course can be taught by many instructors. A student can enroll for any number of courses. Each course can have any number of students, but the course is offered if and only if at least five students are enrolled in the course.

### Exercise #7 Hotel management system

Prepare a class diagram that shows the relationships between each object in a hotel management system. The major information to be maintained in system are as under:

- guest information

- staff (manager, receptionist)
- room occupancy
- food
- billing

### **Exercise #8 Newspaper Distributor System**

The newspaper distributor daily collects various newspapers from different printing presses. The newspapers are distributed area-wise among the persons who look after that area. The area distributor has many employees who distribute the newspaper to the clients. The area distributor sorts newspapers client-wise and hand over them to his employees. Employees distribute bunch of newspapers to each client. On demand of his client, the newspaper distributor also distributes periodicals. In the month end, he delivers bill to each client.

### **Exercise #9 Hospital Management System**

A patient can make many appointments with one or more doctors in the clinic, and a doctor can accept appointments with many patients. However, each appointment is made with only one doctor, and each appointment references a single patient.

- Emergency cases don't require an appointment. However, for appointment management purposes, an emergency is entered in the appointment book as "unscheduled".
- If kept, an appointment yields a visit with the doctor specified in the appointment. The visit yields a diagnosis and, when appropriate, treatment.
- With each visit, the patient's records are updated to provide a medical history.
- Each patient visit creates a bill. Each patient visit is billed by one doctor, and each doctor can bill many patients.

### **Exercise #10 Online Shopping System**

Prepare a class diagram which shows a model for online shopping system. Prepare a diagram that shows the relationships between each objects listed below:

- Customer, Web User, Account, Shopping Cart, Product, Order, Payment, etc.

Each customer has unique id and is linked to exactly one account. Account owns shopping cart and orders. Customer could register as a web user to be able to buy items online. Customer is not required to be a web user because purchases could also be made by phone or by ordering from catalogues. Web user has login name which also serves as unique id. Web user could be in several states - new, active, temporary blocked, or banned, and be linked to a shopping cart. Shopping cart belongs to account.

\*\*\*\*\*