Module Test (Practical): Module 1 – Core Java 8
Max Marks: 70
Duration: 150 Minutes
Mode: Open Book

Assessment Parameters

- *Complete flow of the application with validation and exception handling -70%*
- *Comments/best practice, coding standards – 10%*
- *Execution of the application (Output) – 20%*
- *ScreenShot should be submitted along with the solution*
- *The solution(Project) created by the trainee should have the name like AppName_Empid Ex:ABCCorp_675467*
- *Code with compilation errors will not be considered for evaluation*

Mobile Recharge Application

Problem Statement: Mobile Service Provider needs an application for Customer to provide support, so the customer is able to perform various operations like **Account Balance Enquiry** and **Recharge Account**. (**Assumption**: Other functionalities are not in the scope of application).
Consider the HashMap AccountEntry for storing MobileNo as Key and Account Object as Value. Perform the operations on this Map.

Display menu to Customer Care Representative to perform various operations

1) Account Balance Enquiry
2) Recharge Account
3) Exit

1. Account Balance Enquiry Option [28 Marks]

If customer selects this option, **MobileNo** should be accepted from the user. If **mobileNo** exists in the **AccountEntry Map,** Balance amount should be displayed, otherwise error message should be displayed.

Sample Run

```
Enter Mobile No : 9922943943

Your Current Balance is Rs. 500.00
```

```
OR

Enter MobileNo : 9834391234

ERROR:  Given Account Id Does Not Exits
```

- **Mark Distribution:**

Getting the Account Balance with proper use of service + DAO layers	10
Displaying Account Details	4
Display Error message if mobile no does not exist	5
Mobile no should be 10 digit	3s
Comments and best practices/standards	3
Proper Exception handling + Resource utilization	3

2. Recharge Account

[35 Marks]

If customer selects this option, then Mobile No and Recharge Amount needs to be entered by the customer.

Depending on the given Mobile no update the final amount in Account table i.e.

Note : New Account Balance=Existing Balance Amount+ New Recharge Amount

On Success Recharge display "Account Recharged successfully : " and also display the account details as below.

Sample Run

```
Enter MobileNo : 9932012345
Enter Recharge Amount : 140

Your Account Recharged Successfully
Hello Manish ,Available Balance is 440 .
OR

Enter Mobile No : 90110212

ERROR:  Cannot Recharge Account as Given Mobile No  Does Not Exits
```

- **Mark Distribution:**

Updating the Account properly with updated balance (Service + DAO+Bean+ calculation of amount)	20
Validating inputs(mobile no and recharge amount) & displaying Error Message	8
Comments and best practices/standards	2
Proper Exception handling + Resource utilization	5

3) **Exit**
If customer opts for Exit, customer should be able to quit from application.

4) Write JUNIT test cases for the rechargeAccount method **[7 Marks]**

Classes to be created

```
com.cg.mra.ui // package containing  main class
public class MainUI{
public static void main(// to display the menu and accept the details from user
// create object for service class and execute the respective methods
}
```

```
com.cg.mra.beans //package containing all beans

public class Account{
        private String accountType;
        private String customerName;
        private double accountBalance;
        // getter, setter & constructor
        }
```

```
com.cg.mra.service // package with service layer class

public interface AccountService {
     Account getAccountDetails(String mobileNo);
     int rechargeAccount(String mobileno, double rechargeAmount);
}
public class AccountServiceImpl implements AccountService {
     Account getAccountDetails(String mobileno){..}
     int rechargeAccount(String mobileNo, double rechargeAmount){..}
}
com.cg.mra.dao-// package containing data access class to perform DAO operations

public interface AccountDao{
    Account getAccountDetails(String mobileNo);
```

```
        int rechargeAccount(String mobileNo, double rechargeAmount);
    }
    public class AccountDaoImpl implements AccountDao {
    Map<String,Account> accountEntry;

        public AccountDAOImpl(){
                accountEntry= new HashMap<>();
                accountEntry.put("9010210131", new Account("Prepaid", "Vaishali", 200));
                accountEntry.put("9823920123", new Account("Prepaid", "Megha", 453));
                accountEntry.put("9932012345", new Account("Prepaid", "Vikas", 631));
                accountEntry.put("9010210131", new Account("Prepaid", "Anju", 521));
                accountEntry.put("9010210131", new Account("Prepaid", "Tushar", 632));
        }
    Account getAccountDetails(String mobileNo);
    int rechargeAccount(String mobileNo, double rechargeAmount);
}
```

Add appropriate user defined exception classes and any other supporting classes required.