Day 1:

Topics Covered: data types, operators, if else constructs

Assignment 1:
You are assigned to develop a module to calculate the electricity bill based on below conditions:
The standard price per unit is Rs. 1.20.

- If number of units are less than 100 then standard price per unit will be applied.
- If it is less than or equal to 300 units then Rs. 2 will be charged for number of units over and above 100 units.
- If it is greater 300 units then Rs. 2 will charged for additional 200 units above 100 units and Rs. 3 will be charged for additional units above 300.

Topics Covered: loops and arrays

Assignment 2:
As a developer, you are assigned to develop a module to generate innings statistics of a batsman. Assume the batsman has played 5 overs (30 balls). Generate random runs between 1 to 6 and calculate the below stats:

1. Total runs scored.
2. Number of 0s, 1s, 2s, 3s, 4s and 6s.
3. Strike Rate (runs per ball).

Assignment 3:
Extend the assignment 3, which fetch details for last 5 innings and calculate the following:

1. Average score of last 5 matches
2. Total runs
3. Number of 0s, 1s, 2s, 3s, 4s and 6s.
4. Average Strike Rate (runs per ball).

Topics covered: Classes, objects, constructors, getter & setters.

Assignment 5:
As a developer, you are asked to create a module to store details of a bank account. You are asked to create a class Account with following fields:

- accountNo
- accountBalance
- accountPassword

In addition to above fields, declare a class variable “bankName” to be shared by all objects of the class.
For security reasons, above fields must not be directly accessed outside the class. You need to generate getter and setter methods to let other classes access or modify the object’s details.

Write default and parameterized constructors to allow creation of object in flexible manner.
Write a member method called “displayAccount” in the Account class.
The “displayAccount” method which displays all the details of the account.

Define a main class with “main” where account object is created and call the display method to display account details.

Day 2

Topics Covered: Inheritance

Assignment 1:
Extend the Account class from the Day1-Assignment 5, to create two more classes as “SavingsAccount” and “CurrentAccount”.
Additional fields for SavingsAccount class will be minimumBalance and for CurrentAccount, it will be overdraftLimitAmount.

Override the displayAccount method to display account details along with additional fields of Savings and Current Account.

Topics Covered: Abstract classes and methods

Assignment 2:
Modify the “Account” class to make it an abstract. Define an abstract method “withdraw” which accepts a parameter “amount” to withdraw and display the balance amount.

Override the withdraw method in the SavingsAccount and CurrentAccount class to display account balance and overdraftLimit left respectively.

Topics Covered: Interface
Assignment 3:
Create an interface “ATM” which contains following abstract methods:

- withdraw(int accountNumber,double amount)
- changePassword(int accountNumber,String oldPassword,String newPassword)
- checkBalance()

Create two implementation classes and override the methods of the ATM interface as follows:

- SbiAtm
- IciciAtm

Day 3:

Topics Covered: Exception handling

Assignment 1:
Create a custom exception class “InvalidAmountException” class to throw an exception for negative amount.

Create another custom exception class “InsufficientFundException” class to throw an exception for amount exceeding the current balance.

Topics Covered: Collections

Assignment 2:

Follow the below steps to add a DAO Layer:

1. Create an interface AccountDao with below methods:
    - addAnAccount(Account account)
    - withdraw(int accountNumber,double amount)
    - checkBalance()
    - changePassword(int accountNumber,String oldPassword,String newPassword)
    - List<Account> viewAllAccounts()
    - getAccountDetails(int accountNumber)

2. Create an Implementation class “InMemoryAccountDaoImpl” to implement above methods using static ArrayList collection as data store.

Day 4

Topics Covered: JDBC

Assignment 1:
Create an implementation class “AccountDaoImpl” to implement the AccountDao interface of Assignment2-Day3 using JDBC API and Oracle database.