

# Assignment Day5:

## Question 1:

### 1. Employee Salary Calculation

Create a class Employee as Java Bean class with the following private member variables.

- int employeeId
- String employeeName
- double salary
- double netSalary

Write the following method in the Employee class: **public void calculateNetSalary(int pfpercentage)** - This method should take PF percentage as argument. Deduct the PF amount from the salary and set the netSalary.

Create a Main class which has the main method which invokes the method to get the input and prints the details as shown in the sample.

Also write a method :

public static Employee getEmployeeDetails() - which gets the employee details - id, name and salary from the user and returns the employee object.

public static int getPFPercentage() - which gets the PF percentage and returns the same

In the main method invoke the above two methods, and then call the calculateNetSalary method in Employee class and print the output as shown below.

#### Sample Input 1:

Enter Id: 101 Enter Name: Vivek Enter salary: 20000 Enter PF percentage: 7

#### Sample Output 1:

Id : 101

Name : Vivek

Salary : 20000.0

Net Salary : 18600.0

## Question 2: Ticket Price Calculation

Create a class Ticket with the following private variables int ticketid; int price; static int availableTickets; Include getters and setters methods in the Ticket class.

AvailableTickets should hold only positive value. Zero and negative values are not allowed.(This logic should be checked inside the corresponding setter method)

Write the following method in the Ticket class:

public int calculateTicketCost(int nooftickets) —this method should check the ticket availability, If the tickets are available, reduce the nooftickets from availableTickets and calculate the total amount as nooftickets\*price and return the total amount. If the tickets are not available, this method should return -1.

Write a main method in the Main class to test the application

### Sample input and output

Enter no of bookings: 2 Enter the available tickets: 25 Enter the ticketid: 123 Enter the price: 100 Enter the no of tickets: 5 Available tickets: 25

Total amount:500

Available ticket after booking:20

Enter the ticketid: 124 Enter the price: 100 Enter the no of tickets: 2 Available tickets: 20

Total amount:200

Available ticket after booking:18

## Question 3: BankAccountDetails

In the first round of HR interview for a banking sector, HR decides to make candidates design an application which provides only information on transaction like amount withdrawn with respect to fields given. Develop a program to implement this scenario.

Create a class Account with the private attributes:

- accountId int
- accountType String
- balance int

The method **public boolean withdraw(int)** used to calculate the current balance of the respective account. Before that it should enough balance. If there is enough balance, deduct the amount from the balance and print "Balance amount after withdraw: XXX" and return true. If there is no enough balance, print "Sorry!!! No enough balance" and return false.

Create a class AccountDetails with main function and the below methods :

- public Account getAccountDetails() - This methods gets the input related to Account from the user and returns the Account object with all values set. If the input given for balance is less than or equal to zero, consider it as invalid and display "Balance should be positive". Continue this kind of evaluation till user enters a positive value.
- public int getWithdrawAmount() - This methods gets the amount to be withdrawn as input from the user and returns the same. If the input given for amount is less than or equal to zero, consider it as invalid and display "Amount should be positive". Continue this kind of evaluation till user enters a positive value.

**Use appropriate getters and setters.**

**Sample input 1:**

Enter account id:

100 Enter account type:

Savings

Enter balance:

10000 Enter amount to be withdrawn:

500

**Sample Output 1:**

Balance amount after withdraw: 9500

**Sample input 2:** Enter account id:

101 Enter account type:

Savings Enter balance: 1000 Enter amount to be withdrawn: 1500

**Sample Output 2:**

Sorry!!! No enough balance

**Sample input 3:**

Enter account id:

100

Enter account type:

Savings

Enter balance:

- 100

Balance should be positive

Enter balance:

5000

Enter amount to be withdrawn: 500

**Sample Output 1:**

Balance amount after withdraw: 4500

## Question 4:

Create a Student Bean class with the following fields:

roll: Integer

name: String

address: String

marks: Integer

Create a Demo class and perform the following operation in the main method:

Take the number as input from the user, How many Student objects need to be created.

Create an array of Students with those numbers.

Initialize all the student objects by taking details from the user.

print all the Student details

print the average of all the Student marks.