Exercise 1:

Create a class called Employee that includes three pieces of information as instance variables

1. First name (type String)
2. Last name (type String)

Your class should have the following methods:
- Constructor that initializes the three instance variables.
- Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0.

Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

Exercise 2:

Create a class called Date that includes three pieces of information as instance variables

1. Month (type int)
2. Day (type int)
3. Year (type int).

Your class should have the following methods:
- Constructor that initializes the three instance variables and assumes that the values provided are correct.
- Provide a set and a get method for each instance variable.
- Provide a method display Date that displays the month, day and year separated by forward slashes (/).

Write a test application named DateTest that demonstrates class Date’s capabilities.
Exercise 3:
Create a class called Invoice that a hardware store might use to represent an invoice for
an item sold at the store. An Invoice should include four pieces of information as instance
variables:
1. Part number (type String)
2. Part description (type String)
3. Quantity of the item being purchased (type int)
4. Price per item (double).

Your class should have the following:
- Constructor that initializes the four instance variables.
- Provide a set and a get method for each instance variable.
- Provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0.
  If the price per item is not positive, it should be set to 0.0.

Write a test application named InvoiceTest that demonstrates class Invoice’s capabilities.
Exercise 4:
A Café sells coffee for SR 5.50 a cup, Tea for SR 3.50 and Donuts for SR 2.25. Write a Java program to compute a customer’s bill. Declare a class Café and use appropriate data types for declaring the following attributes coffee, tea, donut, discount, subTotal and total. Discount is a number between 0-100 and it represents a percentage, coffee, tea and donut represent amount of items ordered.

Class Café should have the following operations:
1) Constructor to initialize the quantities and discount to 0.
2) setters() Methods for the four attributes.
3) calculateSubTotal() to calculate the subtotal of the bill.
4) double calculateTotal() to calculate the total cost of the bill, including the discount and return this total cost.
5) display() to display an itemized bill as follows: (assume discount is 10)

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coffee</td>
<td>3</td>
<td>SR 16.50</td>
</tr>
<tr>
<td>Tea</td>
<td>2</td>
<td>SR 7.00</td>
</tr>
<tr>
<td>Donuts</td>
<td>2</td>
<td>SR 4.50</td>
</tr>
<tr>
<td>Sub total</td>
<td></td>
<td>SR 28.00</td>
</tr>
<tr>
<td>Discount</td>
<td>(%10)</td>
<td>SR 2.80</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>SR 25.20</td>
</tr>
</tbody>
</table>

Do the following:
(1) Declare the class Café in a separate file called Café.java.
(2) Write the main program to test class Café using Class TestCafe.java. You should read, calculate and display bills for several customers using a menu driven program (Hint: use while loop for the menu). Your program should display a menu with 2 options:
   1) Read, calculate and display bill for customer
   2) Quit
(3) When the user enters 2 for Quit print the total sales for all the operations.