

Assignment No. 2

Q.1 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—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable.

In addition, 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.

Q.2 Create a class called Employee that includes three instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Provide a 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, do not set its value. 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.

Q.3 Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a 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 displayDate that displays the month, day and year separated by forward slashes (/).

Write a test application named DateTest that demonstrates class Date's capabilities.

Q.4 (*Target-Heart-Rate Calculator*) While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the American Heart Association (AHA), *the formula for calculating your maximum heart rate in beats per minute is 220*

Assignment No. 2

minus your age in years. Your target heart rate is a range that's 50–85% of your maximum heart rate. [Note: These formulas are estimates provided by the AHA. Maximum and target heart rates may vary based on the health, fitness and gender of the individual. Always consult a physician or qualified health care professional before beginning or modifying an exercise program.]

Create a class called `HeartRates`. The class attributes should include the person's first name, last name and date of birth (consisting of separate attributes for the month, day and year of birth). Your class should have a constructor that receives this data as parameters. For each attribute provide `set` and `get` methods. The class also should include a method that calculates and returns the person's age (in years), a method that calculates and returns the person's maximum heart rate and a method that calculates and returns the person's target heart rate. Write a Java application that prompts for the person's information, instantiates an object of class `HeartRates` and prints the information from that object—including the person's first name, last name and date of birth—then calculates and prints the person's age in (years), maximum heart rate and target-heart-rate range.

Q.5 A bank accepts fixed deposit for one year or more and the policy it adopts on interest is as follows:

- i) If a deposit is less than Rs.2000 and for 2 or more years, the interest rate is 5% compound annually.
- ii) If a deposit Rs.2000 or more but less than Rs.6000 and for 2 or more years, the interest rate is 7% compound annually.
- iii) If a deposit is more than Rs.6000 and for 1 or more years, the interest rate is 8% compound annually.
- iv) On all deposits for 5 year or more, interest is 10% compound annually.
- v) On all other deposits not covered by above conditions, the interest is 3% compound annually.

Note: Compound Interest = $p \cdot (1 + r/100)^T$ to the power T.
(Hint: Use `java.lang.Math` Class)

Assignment No. 2

Give the amount deposited and the number of years, write a program to calculate the money in the customer's account at the end of the specified time.

Q.6 Create Java application for the Cell phone charges bill: Accept the user's plan and Number of calls made and Number of SMS sent in a month.

| Plan | Monthly Charges | Call Charges | SMS Charges | Free calls | 50% Discount |
|--------|-----------------|--------------|-------------|-----------------|----------------|
| PLAN-A | 199 | 1.00 | 1.00 | First 50 calls | Next 50 calls |
| PLAN-B | 299 | 0.80 | 0.75 | First 75 calls | Next 75 calls |
| PLAN-C | 399 | 0.60 | 0.50 | First 100 calls | Next 100 calls |

For Example: If a person subscribes for PLAN-B and done 300 calls and sent 45 SMS, then Bill should be calculated as follows:

| Items | Amount | Explanation |
|---------------------|------------|--|
| First 75 Calls | Rs. 0 | [(Free calls)] |
| Next 75 Calls | Rs. 30 | [i.e. 75 * 0.40 (50 % Discount)] |
| Remaining 150 Calls | Rs. 120 | [i.e. 150 * 0.80 (Regular Billing)] |
| SMS Bill | Rs. 22.50 | [i.e. 30 * 0.75 (Regular Billing)] |
| Total Charges | Rs. 471.50 | [i.e. Monthly Charges + Bill Charges] |
| Final Bill | Rs. 530.44 | [i.e. Total Charges + Service Tax 12.5%] |

Q.7 Write an application that inputs from the user the radius of a circle as an integer and prints the circle's diameter, circumference and area using the floating-point value 3.14159 for π (PI).

Math.PI for the value of π (PI). This constant is more precise than the value 3.14159. Class Math is defined in package

Assignment No. 2

java.lang. Classes in that package are imported automatically, so you do not need to import class Math to use it.

Use the following formulas (r is the radius):

$diameter = 2r$

$circumference = 2\pi r$

$area = \pi r^2$

Q.8 (Computerization of Health Records)

A health care issue that has been in the news lately is the computerization of health records. This possibility is being approached cautiously because of sensitive privacy and security concerns, among others. Computerizing health records could make it easier for patients to share their health profiles and histories among their various health care professionals. This could improve the quality of health care, help avoid drug conflicts and erroneous drug prescriptions, reduce costs and, in emergencies, could save lives.

In this exercise, you'll design a HealthProfile class for a person.

The class attributes should include the person's first name, last name, gender, date of birth (consisting of separate attributes for the month, day and year of birth), height (in inches) and weight (in pounds).

Your class should have a constructor that receives this data. For each attribute, provide set and get methods.

The class also should include methods that calculate and return the user's age in years, maximum heart rate and target-heart-rate range, and body mass index.

Write a Java application that prompts for the person's information, instantiates an object of class HealthProfile for that person and prints the information from that object—including the person's first name, last name, gender, date of birth, height and weight—then calculates and prints the person's age in years, BMI, maximum heart rate and target-heart-rate range.

Note : Use previous assignment for calculating BMI and heart rate

Q.9 (Credit Limit Calculator)

Develop a Java application that determines whether any of several department-store customers has exceeded the credit limit on a charge account.

Assignment No. 2

For each customer, the following facts are available:

- a) account number
- b) balance at the beginning of the month
- c) total of all items charged by the customer this month
- d) total of all credits applied to the customer's account this month
- e) allowed credit limit.

The program should input all these facts as integers, calculate the new balance ($= \text{beginning balance} + \text{charges} - \text{credits}$),

display the new balance and determine whether the new balance exceeds the customer's credit limit. For those customers whose credit limit is exceeded, the program should display the message "Credit limit exceeded".

Q.10 (Sales Commission Calculator)

A large company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9% of their gross sales for that week.

For example, a salesperson who sells \$5000 worth of merchandise in a week receives \$200 plus 9% of \$5000, or a total of \$650. You've been supplied with a list of the items sold by each salesperson. The values of these items are as follows:

| Item | Value |
|------|--------|
| 1 | 239.99 |
| 2 | 129.75 |
| 3 | 99.95 |
| 4 | 350.89 |

Develop a Java application that inputs one salesperson's items sold for last week and calculates and displays that salesperson's earnings. There's no limit to the number of items that can be sold.