JAVA PROGRAMMING COURSE (SWE2023)
SPRING SEMESTER 2021

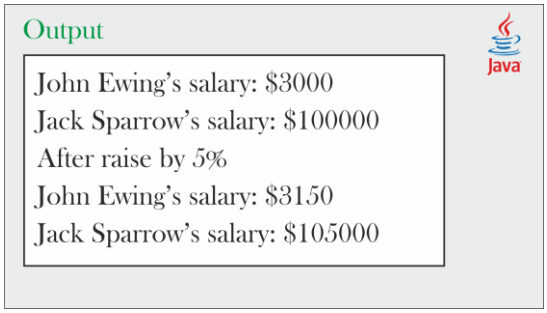
INSTRUCTOR: Prof. TAMER ABUHMED
COLLEGE OF SOFTWARE

Assignment 2

This assignment consists of three tasks. Guidelines for submission format is given at the end of the assignment file.

Task 1

(Employee Class) 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 app named *EmployeeTest* that demonstrates class Employee's capabilities. Create two Employee objects and display each object's *monthly* salary. Then give each Employee a 5% raise and display each Employee's monthly salary again.

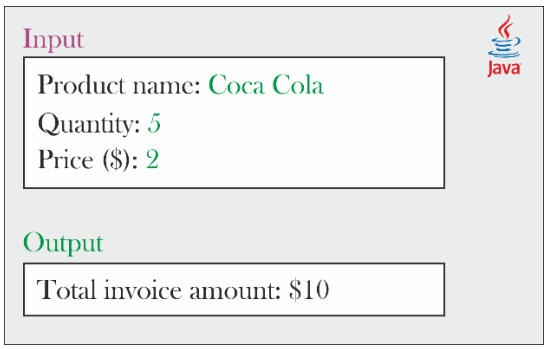


Task 2

(Invoice Class) Create a class Invoice for POS (Point of Sale) system that calculates total cost of the shopping. Invoice class includes three instance variables – product name (type String), quantity (type Integer) and price (type Double). Then, provide a method named ***getInvoiceAmount*** that calculates the invoice amount.

Use constructor to initialize those three instance variables. Provide a *set* and a *get* method for each instance variable. In this task, use class Scanner to input the data. Then, calculate the invoice amount and print it.

Write a test app named *InvoiceTest* that demonstrates class Invoice's capabilities.



Task 3

(BankAccount Class) Create a class called BankAccount that includes five instance variables –a first name (type String), last name (type String) and balance (type Double). Use constructor to initialize those three instance variables. Provide a *set* and a *get* method for each instance variable. Provide following methods:

- *Withdraw (double amount)* – you can withdraw money from account using this method. Ensure that the withdrawal amount does not exceed the BankAccount's balance. If it does, the balance should be left unchanged and the method should print a message indicating "Withdrawal amount exceeded account balance."
- *Transfer (BankAccount receiverObject, double amount)* – sender can transfer money from his account to other account using this method. Ensure that the sender amount does not exceed the BankAccount's balance. If it does, the balance should be left unchanged and the method should print a message indicating, "" After transfer, the amount of money should be added to the receiver's account balance.

Write a test app named *BankAccountTest* that demonstrates class BankAccount's capabilities. Create two BankAccount objects and try following operations:

Output

```
Balance ->
John Ewing's balance: $14000
Jack Sparrow's balance: $100000
Operation ->
John Ewing withdrew 2000$
Success.
Balance ->
John Ewing's balance: $12000
Jack Sparrow's balance: $100000
Operation ->
Jack Sparrow transferred $50000 to John Ewing
Success.
Balance ->
John Ewing's balance: $62000
Jack Sparrow's balance: $50000
Operation ->
Jack Sparrow withdrew $60000
Failed. Withdrawal amount exceeded account balance.
Balance ->
John Ewing's balance: $62000
Jack Sparrow's balance: $50000
Operation ->
Jack Sparrow transferred $55000 to John Ewing
Failed. You have not enough money!
Balance ->
John Ewing's balance: $62000
Jack Sparrow's balance: $50000
```

Submission format: Submit three separate files. Files must include the implementation code of each task and comments for important lines of code to explain the purpose. All the files should be submitted as a **zip** file.

Name of zip file: {student ID}_{Student name}_assignment2.zip
Example: 2020712837_Frank_Thomas_assignment2.zip

NOTE: Plagiarism is strictly prohibited. If there is any plagiarism found in the code, you will be given an "F" for the assignment.

If you have any questions about the assignment 2, you can ask TAs (Firuz Juraev – f.i.juraev@gmail.com).

Good luck!