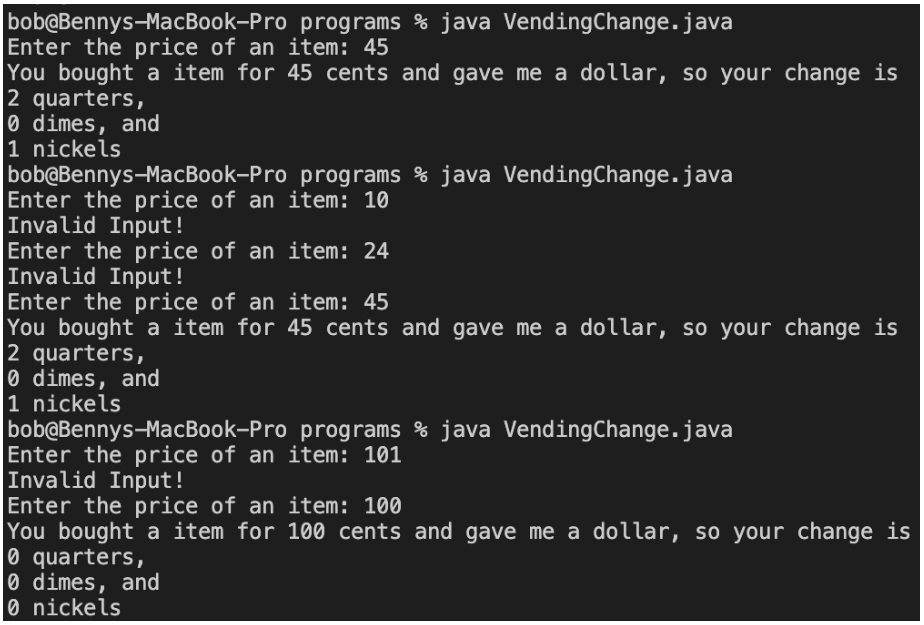
Homework 3 Report

Benny Chen

February 16, 2023

Problem A

Upgrade the program VendingChange.java from Homework #1 to prompt the user to enter the input price. The program then checks whether the input price entered by the user conforms to the specifications. Recall that the two constraints on the input price are: (i) it should be more than 25c but less than 100c; both inclusive and (ii) it should be a multiple of 5. If either of these conditions is not met, the program should print a message “Invalid Input” and prompt the user to enter a new input. This process must be repeated until the user provides a valid price.



```
bob@Bennys-MacBook-Pro programs % java VendingChange.java
Enter the price of an item: 45
You bought a item for 45 cents and gave me a dollar, so your change is
2 quarters,
0 dimes, and
1 nickels
bob@Bennys-MacBook-Pro programs % java VendingChange.java
Enter the price of an item: 10
Invalid Input!
Enter the price of an item: 24
Invalid Input!
Enter the price of an item: 45
You bought a item for 45 cents and gave me a dollar, so your change is
2 quarters,
0 dimes, and
1 nickels
bob@Bennys-MacBook-Pro programs % java VendingChange.java
Enter the price of an item: 101
Invalid Input!
Enter the price of an item: 100
You bought a item for 100 cents and gave me a dollar, so your change is
0 quarters,
0 dimes, and
0 nickels
```

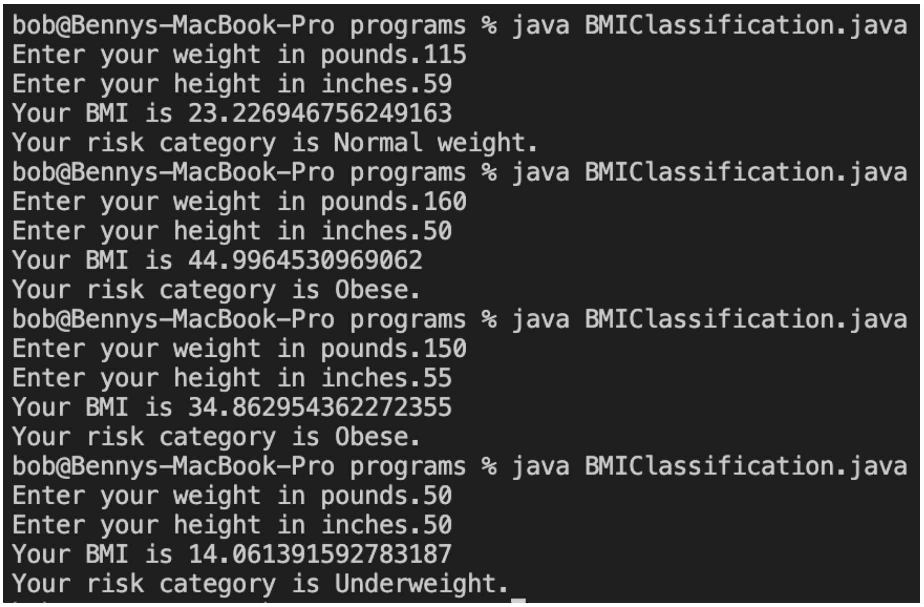
Problem B

Write a program BMIClassification.java that prompts the user to input their weight (in pounds) and height (in inches). Note that both the weight and height can be real. The program then converts the weight to kilograms and height to meters, and calculates the BMI according to the equation:

$$BMI = weight/(height^2)$$

The program prints the BMI of the user. Further, based on the value of the BMI, the program produces the risk classification of the user according to the following rules.

- Underweight – less than 18.5
- Normal weight – greater than or equal to 18.5 and less than 25
- Overweight – greater than or equal to 25 and less than 30
- Obese – greater than or equal to 30.



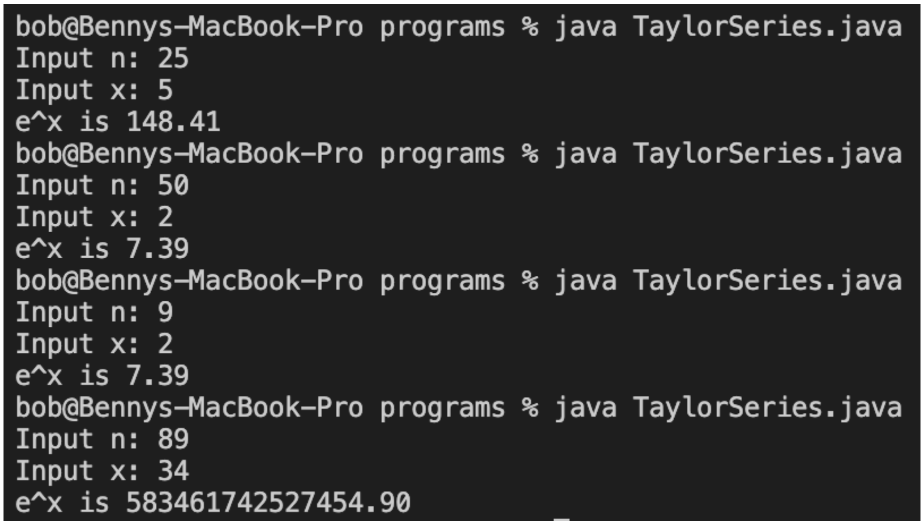
```
bob@Bennys-MacBook-Pro programs % java BMIClassification.java
Enter your weight in pounds.115
Enter your height in inches.59
Your BMI is 23.226946756249163
Your risk category is Normal weight.
bob@Bennys-MacBook-Pro programs % java BMIClassification.java
Enter your weight in pounds.160
Enter your height in inches.50
Your BMI is 44.9964530969062
Your risk category is Obese.
bob@Bennys-MacBook-Pro programs % java BMIClassification.java
Enter your weight in pounds.150
Enter your height in inches.55
Your BMI is 34.862954362272355
Your risk category is Obese.
bob@Bennys-MacBook-Pro programs % java BMIClassification.java
Enter your weight in pounds.50
Enter your height in inches.50
Your BMI is 14.061391592783187
Your risk category is Underweight.
```

Problem C

Write a program TaylorSeries.java that calculates ex as a sum of the first n terms of the following Taylor series:

$$e^x = 1 + x + \ldots + \frac{x^n}{n!}$$

The program should prompt the user to input n and x, and print ex as output, accurate up to two decimal places. n is an integer and x is a real number.



```
bob@Bennys-MacBook-Pro programs % java TaylorSeries.java
Input n: 25
Input x: 5
e^x is 148.41
bob@Bennys-MacBook-Pro programs % java TaylorSeries.java
Input n: 50
Input x: 2
e^x is 7.39
bob@Bennys-MacBook-Pro programs % java TaylorSeries.java
Input n: 9
Input x: 2
e^x is 7.39
bob@Bennys-MacBook-Pro programs % java TaylorSeries.java
Input n: 89
Input x: 34
e^x is 583461742527454.90
```

Problem D

The transactions at a store are saved in a txt file with the following pre-specified format:
SKU,Quantity,Price,Description

- 4039,50,0.99,SODA
- 9100,5,9.50,T-SHIRT
- 1949,30,110.00,JAVA PROGRAMMING TEXTBOOK
- 5199,25,1.50,COOKIE

Write a program TransactionReport.java that prompts the user for the name of the input file, reads the transactions from the file, after skipping over the first line which is the header. The program then produces a transaction report by first computing the sale amount for each item as a product of the quantity and price, and then computing the total sale across all the items. Thus, the output produced by processing the above file is as follows.

Test Cases

Trans.txt:

- SKU,Quantity,Price,Description
- 4039,50,0.99,SODA

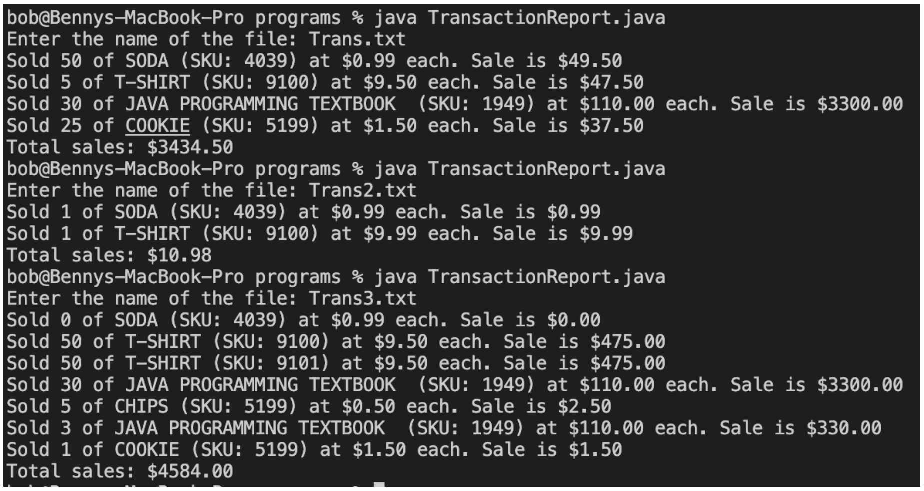
- 9100,5,9.50,T-SHIRT
- 1949,30,110.00,JAVA PROGRAMMING TEXTBOOK
- 5199,25,1.50,COOKIE

Trans2.txt:

- SKU,Quantity,Price,Description
- 4039,1,0.99,SODA
- 9100,1,9.99,T-SHIRT

Trans3.txt:

- SKU,Quantity,Price,Description
- 4039,0,0.99,SODA
- 9100,50,9.50,T-SHIRT
- 9101,50,9.50,T-SHIRT
- 1949,30,110.00,JAVA PROGRAMMING TEXTBOOK
- 5199,5,.50,CHIPS
- 1949,3,110.00,JAVA PROGRAMMING TEXTBOOK
- 5199,1,1.50,COOKIE



```
bob@Bennys-MacBook-Pro programs % java TransactionReport.java
Enter the name of the file: Trans.txt
Sold 50 of SODA (SKU: 4039) at $0.99 each. Sale is $49.50
Sold 5 of T-SHIRT (SKU: 9100) at $9.50 each. Sale is $47.50
Sold 30 of JAVA PROGRAMMING TEXTBOOK  (SKU: 1949) at $110.00 each. Sale is $3300.00
Sold 25 of COOKIE (SKU: 5199) at $1.50 each. Sale is $37.50
Total sales: $3434.50
bob@Bennys-MacBook-Pro programs % java TransactionReport.java
Enter the name of the file: Trans2.txt
Sold 1 of SODA (SKU: 4039) at $0.99 each. Sale is $0.99
Sold 1 of T-SHIRT (SKU: 9100) at $9.99 each. Sale is $9.99
Total sales: $10.98
bob@Bennys-MacBook-Pro programs % java TransactionReport.java
Enter the name of the file: Trans3.txt
Sold 0 of SODA (SKU: 4039) at $0.99 each. Sale is $0.00
Sold 50 of T-SHIRT (SKU: 9100) at $9.50 each. Sale is $475.00
Sold 50 of T-SHIRT (SKU: 9101) at $9.50 each. Sale is $475.00
Sold 30 of JAVA PROGRAMMING TEXTBOOK  (SKU: 1949) at $110.00 each. Sale is $3300.00
Sold 5 of CHIPS (SKU: 5199) at $0.50 each. Sale is $2.50
Sold 3 of JAVA PROGRAMMING TEXTBOOK  (SKU: 1949) at $110.00 each. Sale is $330.00
Sold 1 of COOKIE (SKU: 5199) at $1.50 each. Sale is $1.50
Total sales: $4584.00
```