

**CSE 2102: Introduction to Software Engineering**  
**Conditionals, Branching, and Arrays**  
**Assigned: February 9, 2023, Due: February 23, 2023**

**Problem A (25 pts.)**

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.

Input cmd:

```
java VendingChange
```

Output:

**First Test Case:**

```
Enter the price of an item: 45
You bought an item for 45 and gave me a dollar,
so your change is
2 quarters,
0 dimes, and
1 nickel
```

**Second Test Case:**

```
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 an item for 45 and gave me a dollar,
so your change is
2 quarters,
0 dimes, and
1 nickel
```

### Third Test Case:

```
Enter the price of an item: 101
Invalid Input!
Enter the price of an item: 100
You bought an item for 100 and gave me a dollar,
so your change is
0 quarters,
0 dimes, and
0 nickel
```

### Problem B (25 pts.)

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.

**Input cmd:**

```
java BMIClassification
```

**Output:**

```
Enter your weight in pounds.115
Enter your height in inches.59
Your BMI is 23.275751025138153
Your risk category is Normal weight.
```

### Problem C (25 pts.)

Write a program `TaylorSeries.java` that calculates  $e^x$  as a sum of the first  $n$  terms of the following Taylor series:

$$e^x = 1 + x + \dots + x^n/n!$$

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

Input cmd:

```
java TaylorSeries
```

Output:

```
Input n: 25
Input x: 5
e^x is: 148.41
```

### **Problem D (25 pts.)**

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.

Input cmd:

```
java TransactionReport
```

Output:

```
Please enter the transaction filename: Transactions.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
```

### **Submission**

The following deliverables must be submitted on HuskyCT by midnight on February 23, 2023.

- Java and class files, Java code must be well-documented.
- For Problems A-C, two test cases should be provided as screenshots. For Problem D, two input files used as test cases and their corresponding outputs should be provided.
- Please make sure that your code compiles, we will test your code offline with specific test cases (common to all).
- Late submissions (without any legitimate excuse) will incur a penalty of 10% per day.