CS1341 – Lab #3 Due Saturday 02/26/2022 at 6:00 am

Pre-Lab

There is no pre-lab this week. Be sure to read the instructions thoroughly *before* your lab class so you can identify questions to make sure you completely understand the lab requirements and the skills required to be able to complete it.

GRADING: Comment your program to explain your steps. Each program should compile without errors and should run to produce outputs described for each exercise. The following points will be discounted if the related element is missing or incorrect:

- Output formatting [2 points each]
- Proper names for classes, variables, and methods [1 point each]
- No Comments [5 points]
- Program doesn't compile [5 points for each minor error up to 5 errors provided that after fixing the errors the program compiles. If the program does not compiler after the 5 errors are fixed, partial credit will be given not to exceed 50 points]
- Source code (java file) missing [15 points each]
- Executable (class file) missing [15 points each]
- Both java files and class files missing [90 points]
- Missing method where required [5 points each]
- Missing loop where a loop is required [5 points each]
- Missing if-else where required [5 points each]

Plagiarism or collaboration with anyone other than your professor, lab instructor, or CS help desk personnel will result in no credit for the assignment and possible honor code violation. Plagiarism is inclusion of any line of code that was created by another person, regardless of the source.

Complete the code for each of the lab problems so they compile and run successfully. Submit the java and class files via Canvas (as a single zip-file). Include a comment block at the top of each Java file that includes your name, student id number, and "**Lab 3-Spring-2022**".

LAB Problem 1 [10 points]

Write a Java program called SumSnake. This program should have a single class called SumSnake and a single method (the main method). The program should do the following:

1. Prompt the user to enter an integer number using the keyboard
2. Use a ***for*** loop to count the natural numbers from 0 up to the number they entered. Print one number per line with a space preceding each even number. Calculate and output the sum of these numbers after the "snake" of numbers.
3. Use a ***while*** loop to accomplish the same thing

Hint: Remember that all even numbers are divisible by 2.

A sample output is provided below. In this example, the user enters a single integer, 10.

```
Enter an integer: 10
-----For Loop-----
 0
1
 2
3
 4
5
 6
7
 8
9
Sum is 45
-----While Loop-----
 0
1
 2
3
 4
5
 6
7
 8
9
Sum is 45
```

LAB Problem 2 [45 points]

****Before you write any code, be sure to read ALL the instructions****

Assume you are in charge of inventory management for the SMU Bookstore. The two main products at the bookstore are Textbooks and SMU Branded Apparel. The Bookstore may purchase textbooks for $50 each and then resell them for $100 each. SMU Branded Apparel can be purchased for $5 each and sold for $13.37 each. The Bookstore must have items in stock before it can sell them and also must have enough cash on hand in the Bookstore's bank account to purchase / re-stock new items.

The purpose of this program is to simulate an administrative portal for Bookstore inventory management for the SMU Bookstore. We should be able to view our current items in stock and bank account balance. We should also be able to handle the purchase and sale of items (Books and Apparel in this case). We'll also need to make sure we have enough money in our bank account to purchase items before buying them and enough items in stock to sell before performing a sale.

When the program starts it:

- Sets our current account balance to 0 since we haven't sold anything yet
- Sets our number of books in stock to 100
- Sets our number of apparel in stock to 250
- Displays a menu saying:

```
Welcome to the SMU Bookstore Inventory System
_____________________________

Choice   Action
1        Show Inventory
2        Show Balance
3        Buy Books
4        Buy SMU Apparel
5        Sell Item(s)
6        Exit
Choose an option:
```

Requirements:

- Within a loop, display the menu. If the user enters an option that is not 1, 2, 3, 4, 5 or 6, display the message "Invalid choice. Please choose an option from the list." and loop back around to redisplay the menu. Continue in this manner until a correct option is input.
- If option 1 is selected, display the current inventory with prefix text: "Books in Stock: " and "Apparel in Stock: ".

- If option 2 is selected, display the current account balance with prefix text: “Current Balance: “.
- If option 3 is selected, prompt the user for the number of Books to buy, and once we have this number, check to see if we can afford the Books (based on the formula: number of Books to be bought * 50 <= total cash we have)
    - If we can, then we add the newly purchased Books to our current book stock and reduce our account balance to whatever is left after the purchase. Then we display the number of books bought, followed by our inventory and account balance using the same procedure as options 1 and 2.
    - If we cannot, then we display a message saying – “We don’t have enough money for this :(”.
- If option 4 is selected, prompt the user for the number of Apparel to buy, and once we have this number, check to see if we can afford the Apparel (based on the formula: number of Apparel to be bought * 5 <= total cash we have)
    - If we can, then we add the newly purchased Apparel to our current apparel stock and reduce our account balance to whatever is left after the purchase. Then we display the number of apparel bought, followed by our inventory and account balance using the same procedure as options 1 and 2.
    - If we cannot, then we display a message saying – “We don’t have enough money for this :(”.
- If option 5 is selected, prompt the user for the number of Books to sell. Then prompt the use for the number of Apparel to sell. Once we have these numbers, check to see if we have enough items in stock
    - If we can, then we reduce the stock count by the number of each item sold and increase the money we have (by number of Books sold * 100 + number of Apparel sold * 13.37). Then we display our inventory and account balance using the same procedure as options 1 and 2.
    - If we cannot, then we display a message saying – “Balance too low to make this purchase!”.

- If option 6 is selected, then the program will end. Display a message saying “Goodbye :)”.

A sample output is provided below where user input is highlighted in yellow.

```
Welcome to the SMU Bookstore Inventory System
_____________________________

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 1
Books in Stock:        100
Apparel in Stock:  250


Welcome to the SMU Bookstore Inventory System
```

```
_________________________

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 2
Current Balance:       0.00

Welcome to the SMU Bookstore Inventory System
---------------------------------------------

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 5
Enter number of books to sell: 251
Enter number of apparel to sell: 1
We don't have enough items in stock for this :(

Welcome to the SMU Bookstore Inventory System
_________________________

1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 3
Enter number of books to buy: 1
We don't have enough money for this :(

Welcome to the SMU Bookstore Inventory System
_________________________

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 4
Enter amount of apparel to buy: 2
We don't have enough money for this :(

Welcome to the SMU Bookstore Inventory System
```

```
_________________________

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 5
Enter number of books to sell: 10
Enter number of apparel to sell: 10
Books in Stock:        90
Apparel in Stock:  240
Current Balance:       1133.70

Welcome to the SMU Bookstore Inventory System
_________________________

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 3
Enter number of books to buy: 2
Bought 2 books
Books in Stock:        92
Apparel in Stock:  240
Current Balance:       1033.70

Welcome to the SMU Bookstore Inventory System
_________________________

Choice    Action
1         Show Inventory
2         Show Balance
3         Buy Books
4         Buy SMU Apparel
5         Sell Item(s)
6         Exit
Choose an option: 4
Enter amount of apparel to buy: 6
Bought 6 apparel
Books in Stock:        92
Apparel in Stock:  246
Current Balance:       1003.70

Welcome to the SMU Bookstore Inventory System
_________________________

Choice    Action
1         Show Inventory
```

```
2          Show Balance
3          Buy Books
4          Buy SMU Apparel
5          Sell Item(s)
6          Exit
Choose an option: 9
Invalid choice. Please choose an option from the list.

Welcome to the SMU Bookstore Inventory System
_____________________

Choice     Action
1          Show Inventory
2          Show Balance
3          Buy Books
4          Buy SMU Apparel
5          Sell Item(s)
6          Exit
Choose an option: 1
Books in Stock:        92
Apparel in Stock:  246

Welcome to the SMU Bookstore Inventory System
_____________________

Choice     Action
1          Show Inventory
2          Show Balance
3          Buy Books
4          Buy SMU Apparel
5          Sell Item(s)
6          Exit
Choose an option: 2
Current Balance:   1003.70

Welcome to the SMU Bookstore Inventory System
_____________________

Choice     Action
1          Show Inventory
2          Show Balance
3          Buy Books
4          Buy SMU Apparel
5          Sell Item(s)
6          Exit
Choose an option: 6
Goodbye :)
```

LAB Problem 2 [45 points]

Based on Part I, the Bookstore's business has been booming! For this reason, we must enhance our inventory management program, so it becomes more manageable and expandable. We can do this by changing our single method program into multi methods. Our inventory system still contains basic features, although they are much better designed this time around.

Here's the general structure of the program:

Bookstore2.java

- static variables
- main method
- public static int showMenu()
- public static void showInventory()
- public static void showBalance()
- public static void buyBooks(int numBooks)
- public static void buyApparel(int numApparel)
- public static void sellItems(int numBooks, int numApparel)

First, declare the number of books and apparel we have and account balance to be class member variables. Add the Scanner declaration as well. Here's how:

static int numBooksInStock = 100;

static int numApparelInStock = 250;

static double balance = 0;

static Scanner input = new Scanner(System.in);

Important Note: A variable declared within a class but outside any method is called a *class member variable* or ***field***, in contrast to a local variable defined inside a method. A field's scope extends from the class's opening brace to the class's closing brace and reaches into methods regardless of where the field is declared within the class.

We now want the following methods included:

main – We always need main, but don't want it to do all the work

Note: If a method needs to read user input, a good practice is to create a single Scanner object in main() and pass that Scanner object to the method.

showMenu – Will display the menu and return the number the user selects.

showInventory– Move the logic you created in Part I for option 1 to this method. This method will display the number of Books and Apparel in stock.

showBalance — Move the logic created in Part I for option 2 to this method. This method will display the current balance.

buyBooks – Move the logic you created in Part I for option 3 to this method. The current stock and current balance will be displayed by calling the methods showInventory() and showBalance().

buyApparel – Move the logic you created in Part I for option 4 to this method. The current stock and current balance will be displayed by calling the methods showInventory() and showBalance().

sellItems – Move the logic you created in Part I for option 5 to this method. The current stock and current balance will be displayed by calling the methods showInventory() and showBalance().

If/When main receives the number 6 from the showMenu method, the looping structure within main should stop and the program will end. Display a message saying "Goodbye :)"

Your output should remain the same as in Part I.