**Assignment 3 – BCS 345 Java Programming**

**Due: 10/14/2019 @ 10:50am**

SUBMIT ALL PROJECTS NECESSARY TO RUN THIS PROGRAM. When you submit the assignment on Blackboard make sure your zip file is attached to the submission.

IMPORTANT – Make sure you ***properly comment*** AND ***properly indent*** your program. The commenting and indenting documents are on Blackboard. ***If you fail to properly comment or properly indent I will deduct points.***

***VERY IMPORTANT – IF THE PROGRAM DOES NOT COMPILE THERE WILL BE MAJOR POINTS TAKEN OFF.***

***Overview***

You will be writing the Customer and Purchase classes. There are two main parts to this assignment.

1. Write the Customer and Purchase classes.
2. Write two classes to handle the user interface.

Check the “BCS 345 Eclipse Create Project And Package” document on Blackboard for instructions on how to create projects and packages and how to import them.

***Part 1 - Class Specifications***

**Class – Address (update the existing class)**

Update the Address class. Change the ReadJSON method so that it takes a FileReader instead of a Scanner. Here is the method header:

void ReadJSON(FileReader fr)

This method should not open the FileReader. It should use the GSON library to read the JSON from the file. Data should then be copied into the member variables. Check the JSON slides for help.

**Class – Product (update the existing class)**

Update the Product class. Change the ReadJSON method so that it takes a FileReader instead of a Scanner. Here is the method header:

void ReadJSON(FileReader fr)

This method should not open the FileReader. It should use the GSON library to read the JSON from the file. Data should then be copied into the member variables. Check the JSON slides for help.

**Class – Customer**

Store in project/package: <your last name>.bcs345.purchases.business

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| First | String | Contains the first name of the customer. |
| Last | String | Contains the last name of the customer. |
| Address | Address | Contains the address of the customer. |

***Member Method Signatures and Descirptions (all public)***

|  |  |
| --- | --- |
| **Signature** | **Description** |
| Customer() | Default constructor. Sets the values of each member variable to a default value. Make sure to call new on reference type variables. |
| Get/Set methods | Write get/set methods for all member variables. |
| void Write(PrintStream ps) | Write the contents of all member variables to the given instance of PrintStream. Assume the PrintStream is already open and ready to use. DO NOT ADD ANY DESCRIPTIVE TEXT IN THE OUTPUT. JUST PRINT THE VALUES.  IMPORTANT - Whatever data is written out should be readable by the Read method of this class. If descriptive text is added then Read will not work.  Hint: You can use the Write methods of another class to help with this. |
| void Read(Scanner s) | Read the contents of all member variables from the given instance of Scanner. Assume the following:   1. Scanner is already open. 2. Member variable values are separate lines.   Hint: You can use the Read method of another class to help with this. |
| void WriteJSON(PrintStream ps) | Write the member variables in JSON format to the given PrintStream. |
| void ReadJSON(FileReader fr) | Read the contents of all member variables from the given instance of FileReader as JSON. Assume the following:   1. FileReader is already open. 2. Member variable values are stored in JSON format. |
| @Override  String toString() | This method should show descriptive text and data. It will be used to display data to the user.  Example:  **Jane Doe**  **10 Broadway**  **New York, NY 1001** |

**Class – Purchase**

Store in project/package: <your last name>.bcs345.purchases.business

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| Product | Product | Contains the product being purchased. |
| Quantity | Int | Contains the quantity of product being purchased. |

***Member Method Signatures and Descirptions (all public)***

|  |  |
| --- | --- |
| **Signature** | **Description** |
| Purchase() | Default constructor. Sets the values of each member variable to a default value. Make sure to call new on reference type variables. |
| Get/Set methods | Write get/set methods for all member variables. |
| void Write(PrintStream ps) | Write the contents of all member variables to the given instance of PrintStream. Assume the PrintStream is already open and ready to use. DO NOT ADD ANY DESCRIPTIVE TEXT IN THE OUTPUT. JUST PRINT THE VALUES.  IMPORTANT - Whatever data is written out should be readable by the Read method of this class. If descriptive text is added then Read will not work.  Hint: You can use the Write methods of another class to help with this. |
| void Read(Scanner s) | Read the contents of all member variables from the given instance of Scanner. Assume the following:   1. Scanner is already open. 2. Member variable values are separate lines.   Hint: You can use the Read method of another class to help with this. |
| void WriteJSON(PrintStream ps) | Write the member variables in JSON format to the given PrintStream. |
| void ReadJSON(FileReader fr) | Read the contents of all member variables from the given instance of FileReader as JSON. Assume the following:   1. FileReader is already open. 2. Member variable values are stored in JSON format. |
| @Override  String toString() | This method should show descriptive text and data. It will be used to display data to the user.  Example:  **2, Samsung Galaxy s10, $199.99** |

**Class – CustomerPurchaseConsoleUI**

Store in project/package: <your last name>.bcs345.purchases.presentation

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| No member variables |  |  |

***Member Method Signatures and Descriptions (all public)***

|  |  |
| --- | --- |
| Signature | Description |
| ShowUI() | Shows the user interface. When this method is called it should do the following:   1. Display the menu to the user. 2. Process the user selections   There should be no display or processing code in main. See the Menu Description section below for details. |

**Class – Main (revise from previous assignment)**

Store in project/package: <your last name>.bcs345.purchases.presentation

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| No member variables |  |  |

***Member Method Signatures and Descirptions (all public)***

|  |  |
| --- | --- |
| Signature | Description |
| public static void main(String args[]) | Just create an instance of CustomerPurchaseConsoleUI inside of main and call the ShowUI method on it. No other code should be in main. |

***Customer/Purchase UI Menu Description***

This program will present a menu to the user and then perform an action depending on what the user chooses to do. You should create an instance of Customer inside of ShowUI. When the program runs it should display the menu to the user and give them a chance to input a choice. An action should be taken depending on what choice the user makes. Here is the user menu:

**Customer/Purchase UI**

**--------------------**

**1 – Read customer from file**

**2 – Read customer from file as JSON**

**3 – Write customer to file**

**4 – Write customer to file as JSON**

**5 – Show customer info on screen**

**6 – Read purchase from file**

**7 – Read purchase from file as JSON**

**8 - Write purchase to file**

**9 - Write purchase to file as JSON**

**10 – Show purchase info on screen**

**11 - Exit**

**Enter Choice:**

THE PROGRAM SHOULD KEEP SHOWING THE MENU AND PERFORMING AN ACTION UNTIL THE USER CHOOSES TO EXIT.

***Actions***

|  |  |
| --- | --- |
| Choice | Action |
| 1 | Reads in data for a customer from a user-specified file into the Customer instance. This menu option expects data to come in according to the Customer file format specified at the end of the assignment. The user should be prompted to enter a filename to read from.  Hint: You can use a function on Customer to help out with this. |
| 2 | Similar to option 1 except the input file must be in JSON format. |
| 3 | Writes data from the Customer instance to a file. The user should be prompted to enter a filename to write the data to. Data should be written out according to the Customer file format specified at the end of this assignment.  Hint: You can use a method on Customer to help out with  this. |
| 4 | Similar to option 3 except the output file must be in JSON format. |
| 5 | Show customer information on screen. First, last etc…  Hint: You can use a method on Customer to help out with this. |
| 6 | Reads in data for a purchase from a user-specified file into the Purchase instance. This menu option expects data to come in according to the Purchase file format specified at the end of the assignment. The user should be prompted to enter a filename to read from.  Hint: You can use a function on Purchase to help out with this. |
| 7 | Similar to option 6 except the input file must be in JSON format. |
| 8 | Writes data from the Purchase instance to a file. The user should be prompted to enter a filename to write the data to. Data should be written out according to the Purchase file format specified at the end of this assignment.  Hint: You can use a method on Purchase to help out with  this. |
| 9 | Similar to option 8 except output file must be in JSON format. |
| 10 | Show purchase information on screen. description, price, quantity.  Hint: You can use a method on Purchase to help out with this. |
| 11 | Exit  Do not call System.exit to do this. |

***Deliverables***

***One*** winzip file containing all Eclipse projects necessary to run the program.

***Customer File Format (data on separate lines)***

First

Last

Number

Street

City

State

Zip

***Sample Customer File Data***

**Jane**

**Doe**

**10**

**Broadway**

**New York**

**NY**

**10001**

***Purchase File Format***

Description

Price

Quantity

***Sample Purchase Input File (SamplePurchase.txt)***

**Samsung Galaxy s10**

**199.99**

**2**

***Sample Program Execution***

Customer/Purchase UI

--------------------

1 – Read customer from file

2 – Read customer from file as JSON

3 – Write customer to file

4 – Write customer to file as JSON

5 – Show customer info on screen

6 – Read purchase from file

7 – Read purchase from file as JSON

8 - Write purchase to file

9 - Write purchase to file as JSON

10 – Show purchase info on screen

11 - Exit

Enter Choice: 1

Enter Customer Input Filename: SampleCustomer.txt

Customer/Purchase UI

--------------------

1 – Read customer from file

2 – Read customer from file as JSON

3 – Write customer to file

4 – Write customer to file as JSON

5 – Show customer info on screen

6 – Read purchase from file

7 – Read purchase from file as JSON

8 - Write purchase to file

9 - Write purchase to file as JSON

10 – Show purchase info on screen

11 - Exit

Enter Choice: 5

Jane Doe

10 Broadway

New York, NY 10001

Customer/Purchase UI

--------------------

1 – Read customer from file

2 – Read customer from file as JSON

3 – Write customer to file

4 – Write customer to file as JSON

5 – Show customer info on screen

6 – Read purchase from file

7 – Read purchase from file as JSON

8 - Write purchase to file

9 - Write purchase to file as JSON

10 – Show purchase info on screen

11 - Exit

Enter Choice: 3

Enter Customer JSON Input Filename: CustomerOut.txt

Customer/Purchase UI

--------------------

1 – Read customer from file

2 – Read customer from file as JSON

3 – Write customer to file

4 – Write customer to file as JSON

5 – Show customer info on screen

6 – Read purchase from file

7 – Read purchase from file as JSON

8 - Write purchase to file

9 - Write purchase to file as JSON

10 – Show purchase info on screen

11 - Exit

Enter Choice: 11