**Assignment 2 – BCS 345 Java Programming**

**Due: 9/25/2019 @ 10:50am**

***Submission Guidelines***

Create a winzip file containing the WHOLE project directory and submit on Blackboard.

**READ ALL SPECIFICATIONS.** You only need to create one Java program for this assignment using Eclipse. When you submit the assignment using 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***

There are two main parts to this assignment.

1. Write a project and package that contains definitions for the Address class and the Product class (see below for specifications).
2. Write a project and package that unit tests the Address and Product classes. You will have to import the package that has the definitions of the Address and Product classes.

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.

Write the Address and Product classes. You will also need to write a thorough automated test program. Each of these classes is required to have overrides of the toString method. They should also contain methods to produce JavaScript Object Notation (JSON) strings.

The Address and Product classes should not contain a “main” method. These classes are meant to be part of “libraries” that will be used by other programs.

We are writing two layers of software. There will be a business layer and a presentation layer. The presentation layer will use classes in the business layer.

The business layer will be in one project/package and the presentation layer will be in a different project/package (see deliverables section for exact names of project/packages). The presentation layer project should import the business layer project. Refer to the BCS 345 Eclipse Create Project and Package document on Blackboard to get instruction on how to import a project/package that you create.

The test program (the main method) will be contained in a class in the presentation layer. The test code should ***thoroughly*** test both the Address and Product classes. This means that it should create instances of each class and call ALL methods of each class.

Your final submission should contain **ALL** projects necessary to run the application (all winzipped together).

You need to create input files for this assignment. You should use the data given at the end of the assignment.

***Class Specifications***

**Class – Address**

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

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

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| Number | String | Contains the house number. |
| Street | String | Contains the street name. |
| City | String | Contains the city name. |
| State | String | Contains the state name. |
| Zip | String | Contains the zip code. |

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

|  |  |
| --- | --- |
| **Signature** | **Description** |
| Address() | Default constructor. Sets the values of each member variable to a default value. |
| Address(String number, String street, String city, String state, String zip) | Constructor. Sets the values of each member variable to the corresponding parameter values. |
| 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. |
| 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 on separate lines. |
| void WriteJSON(PrintStream ps) | Write the member variables in JSON format to the given PrintStream.  Example (10 Broadway New York NY 10001):  { "number" : "10", "street" : "Broadway" , "city" : "New York”, "state" : "NY", "zip" : "1001" } |
| void ReadJSON(Scanner s) | Read the contents of all member variables from the given instance of Scanner as JSON. Assume the following:   1. Scanner 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 an address to the user. It should add commas where necessary  For example:  10 Broadway, New York, NY 10001 |

**Class – Product**

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

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

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| Description | String | Contains the product description. |
| Price | double | Contains the price. |

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

|  |  |
| --- | --- |
| **Signature** | **Description** |
| Product() | Default constructor. Sets the values of each member variable to a reasonable default value. |
| 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. |
| 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 on separate lines. |
| void WriteJSON(PrintStream ps) | Write the member variables in JSON format to the given PrintStream.  Example (Samsung Galaxy s10 199.99):  { "description" : "Samsung Galaxy s10", "price" : 199.99 } |
| void ReadJSON(Scanner s) | Read the contents of all member variables from the given instance of Scanner as JSON. Assume the following:   1. Scanner is already open. 2. Member variable values are stored in JSON format. |
| @Override  String toString() | This method should show descriptive text and data. In this case the descriptive text that is added is a comma and a $. The price should show two places after the decimal point.  Example:  Samsung Galaxy s10, $199.99  Hint: You can use the String.format method to help with this. |

**Class – Main**

Store in project/package: <your last name>.bcs345.hwk.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[]) | The automated test code goes in here. See below for description. |

***Automated Test***

**Use the Automated Testing slides on Blackboard as a guide when coding this section.**

Create an automated test in main.

1. **There should be unit tests of ALL get/set methods of both classes.**
2. You need to write code that runs the other methods that are not being unit tested on each class.

The automated test should contain code that demonstrates that ALL methods work on both classes. DO NOT TAKE THE TESTING CODE LIGHTLY. IT IS VERY IMPORTANT TO SHOW THAT EVERYTHING ON EACH CLASS WORKS.

Hint: To truly test if the output produced by the Write method of either class works you need to try and read in the file that it class generates with its Read method. For example, assume you write data to a file named “abc.txt” using the write method. You should now use “abc.txt” file as input to the read method. If the program fails on this read there is an error.

***Deliverables***

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

|  |  |
| --- | --- |
| **Project** | **Description** |
| <your last name>.bcs345.hwk.purchases.business | This project should contain a **package** named <your last name>. bcs345.hwk. purchases.business. Create the **package** under the src directory in the project. This **package** contains the implementation for the program (Java class files). |
| <your last name>.bcs345.hwk.purchases.presentation | This project should contain a **package** named <your last name>. bcs345.hwk. purchases.presentation. Create the **package** under the src directory in the project. This **package** contains the implementation for the program (Java class files).  Contains the test program that imports and uses the business package you created. The test program should be an automated test as described above. |

**Example Project/Package Setup:**

hoskey.bcs345.hwk.purchases.business 🡨 Project

hoskey.bcs345.hwk.purchases.business 🡨 Package

Address.java 🡨 Source file

Product.java 🡨 Source file

hoskey.bcs345.hwk.purchases.presentation 🡨 Project

hoskey.bcs345.hwk.purchases.presentation 🡨 Package

Main.java 🡨 Source file

***Address File Format***

Number

Street

City

State

Zip

***Sample Address Input File (SampleAddress.txt)***

**10**

**Broadway**

**New York**

**NY**

**10001**

***Product File Format***

Description

Price

***Sample Product Input File (SampleProduct.txt)***

**Samsung Galaxy s10**

**199.99**