Contestant Number: _______________

Time: _________

Rank: _________

VISUAL BASIC/ C# PROGRAMMING (330)

NATIONAL – 2019

Knowledge Test:

Certification Exam: MTA Software Development Fundamentals	___________	(100 points)

Production Portion:

Program 1: Player Inventory	______________	(550 points)
TOTAL POINTS	**_____________**	***(650 points)***

Failure to adhere to any of the following rules will result in disqualification:

1. **Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
2. **No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
3. **Electronic devices will be monitored according to ACT standards.**

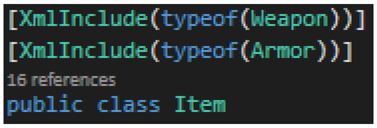
No more than ten (10) minutes orientation
No more than ninety (90) minutes testing time
No more than ten (10) minutes wrap-up

Player Inventory

In this exercise, you will create a Visual Basic or C# Windows Form Application that allows the user to view inventory items and buy from / sell to a shop. Auto-quest allows the player to acquire more gold and the shop can be refreshed to view additional items and replenish gold.

Requirements:

1. You must create a C#/VB Windows Form Application named CS _330_ContestantNumber, where ContestantNumber is your BPA assigned contestant number (including dashes). For example, CS_330_01_2345_6789. If you are using VB, use VB instead of CS in the application name.
2. Your name and contestant number must appear as a comment at the top of the main source code file.
3. Application on Exit button press must ask for confirmation to close, with the Title of the confirmation box being “Confirm” and the text being “Are you sure you wish to exit?”
    a. Exit button must be named “exitButton”
4. The main form must not have minimize or maximize, or close buttons in the Form Handle.
5. The main form must have text “Player Inventory” for its title.
6. A “Item” class must be created containing the following:
    a. Data members string name, double weight, double value, and string image
    b. Constructor for assigning data members and accessors
    c. A virtual function named GetInformationString that returns a string to use for the “itemInfoLabel”.
    d. Hint: In order for Serialization calls to work with “Item” class objects, add the following XmlInclude tags to the Item class definition:



```
[XmlInclude(typeof(Weapon))]
[XmlInclude(typeof(Armor))]
16 references
public class Item
```

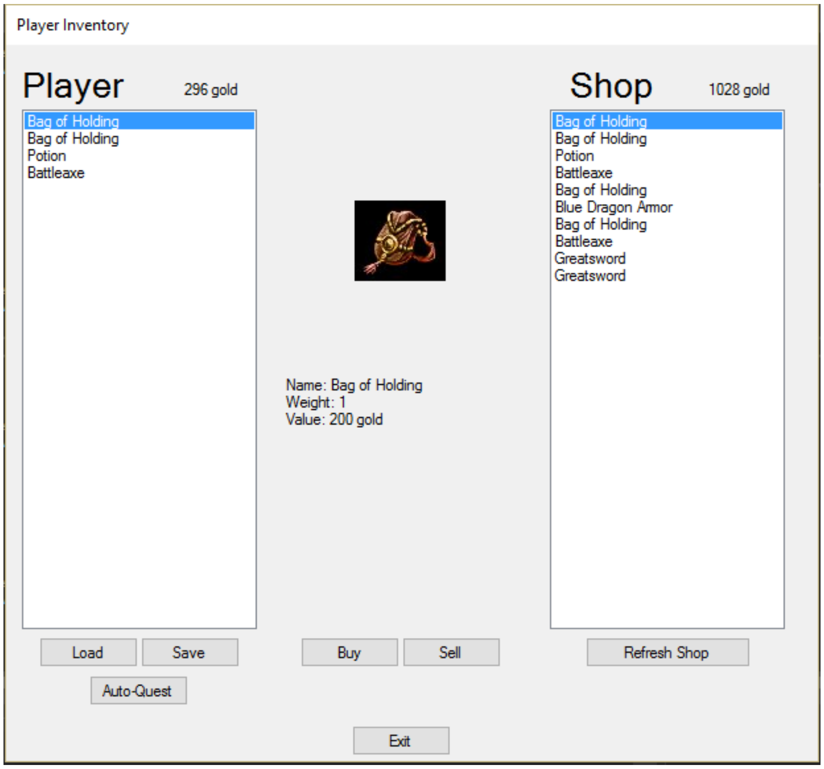
7. A “Weapon” class must be created, inherited from the “Item” class.
    a. Data member string baseDamage is added.
    b. Constructor for assigning data members and accessors.
    c. GetInformationString override that includes the base class information, plus the baseDamage.
8. An “Armor” class must be created, inherited from the “Item” class.
    a. Data member string baseArmor is added.
    b. Constructor for assigning data members and accessors.
    c. GetInformation String override that includes the bass class information, plus the baseArmor.

9. A "Character" class must be created containing the following:
    a. Data members double gold and List<Item> items.
    b. Accessors, BuyItem, and SellItem member functions.

10. On startup, "all_items.xml" must be loaded into global "List<Item> allItems" using the provided "Serialization.cs".
11. On startup, a "Character player" must be populated as a character with four random items from "allItems", 200 base gold, and 0-200 additional gold (selected randomly).
    a. Player items must be shown in a list view on the left side of the screen.
        i. The first item must be selected on startup.
    b. "Player" must be displayed above and aligned to the left of the list box.
    c. "### gold" must be displayed above and aligned to the right of the list box.
12. On startup, a "Character shop" must be populated as a character with ten random items from "allItems", 1000 base gold, and 0-200 additional gold (selected randomly).
    a. Shop items must be shown in a list view on the right side of the screen.
        i. The first item must be selected on startup.
    b. "Shop" must be displayed above and aligned to the right of the list box.
    c. "### gold" must be displayed above and aligned to the right of the list box.
    d. The main form must have a "Refresh Shop" button that repopulates "Character shop" with new random items and gold.
13. The main form must have a "Load" button that allows the user to load a player from file using an OpenFileDialog.
    a. Attempts to load an invalid file must report "Invalid player file selected!" using a MessageBox.
14. The main form must have a "Save" button that allows the user to save a player to file using a SaveFileDialog.
15. The main form must have an "Auto-Quest" button that will give the player a base 20 gold plus an additional 0-20 gold when clicked.
16. The main form must include the image and description of the last selected item from either the player or shop list box.
17. The main form must have a "Buy" button that allows the player to purchase the selected shop item.
    a. If player does not have enough gold, an "Insufficient gold!" MessageDialog must be displayed and the transaction cancelled.
    b. If player does have enough gold, the item is added to the player and removed from the shop. The value of the item is removed from player gold and added to the shop gold.
18. The main form must have a "Sell" button that allows the player to sell the selected player item.
    a. If the shop does not have enough gold, an "Insufficient gold!" MessageDialog must be displayed and the transaction cancelled.

b. If the shop does have enough gold, the item is added to the shop and removed from the player. The value of the item is removed from the shop gold and added to the player gold.

Sample Pictures:



Player 296 gold

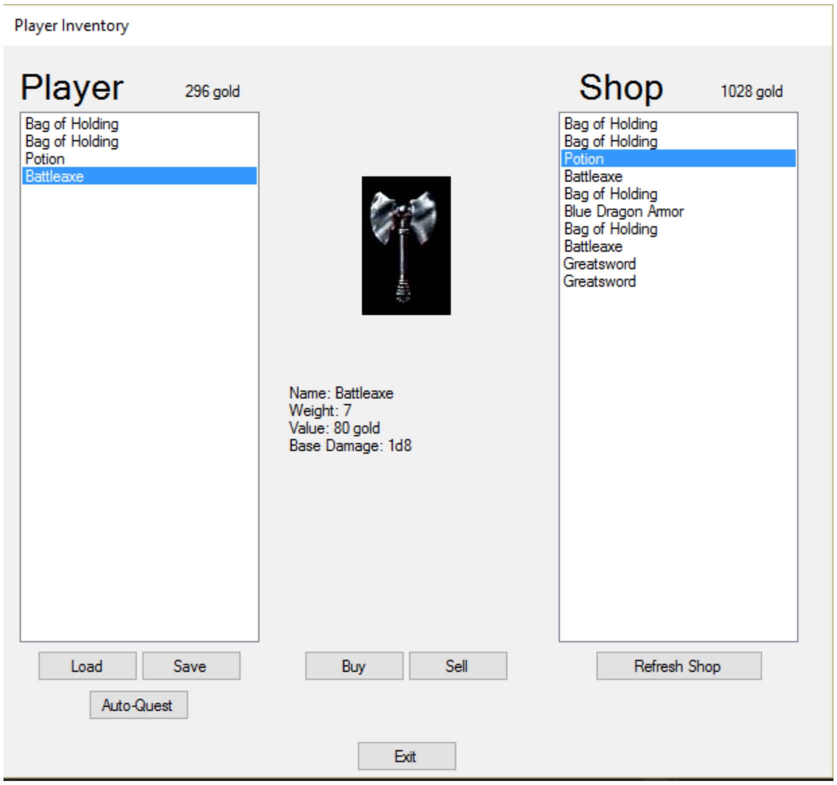
Bag of Holding
Bag of Holding
Potion
Battleaxe

Shop 1028 gold

Bag of Holding
Bag of Holding
Potion
Battleaxe
Bag of Holding
Blue Dragon Armor
Bag of Holding
Battleaxe
Greatsword
Greatsword

Name: Bag of Holding
Weight: 1
Value: 200 gold

Figure 1: Main screen after form loads.



Player 296 gold

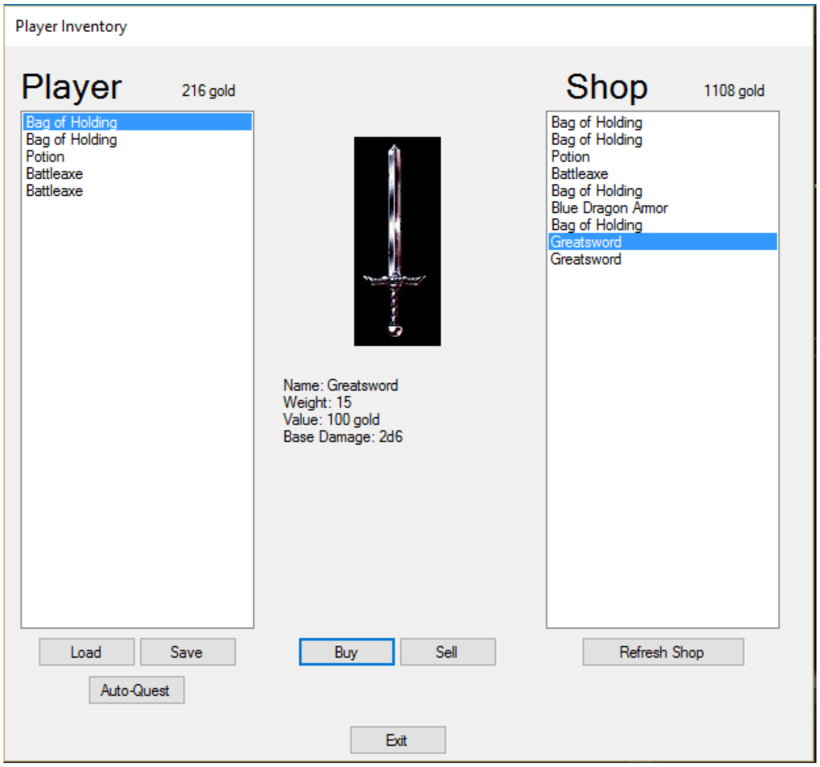
Bag of Holding
Bag of Holding
Potion
Battleaxe

Shop 1028 gold

Bag of Holding
Bag of Holding
Potion
Battleaxe
Bag of Holding
Blue Dragon Armor
Bag of Holding
Battleaxe
Greatsword
Greatsword

Name: Battleaxe
Weight: 7
Value: 80 gold
Base Damage: 1d8

Figure 2: Item information in the center is the last selected item from either player or shop.



Player 216 gold

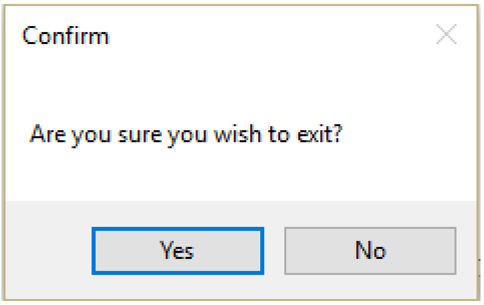
Bag of Holding
Bag of Holding
Potion
Battleaxe
Battleaxe

Shop 1108 gold

Bag of Holding
Bag of Holding
Potion
Battleaxe
Bag of Holding
Blue Dragon Armor
Bag of Holding
Greatsword
Greatsword

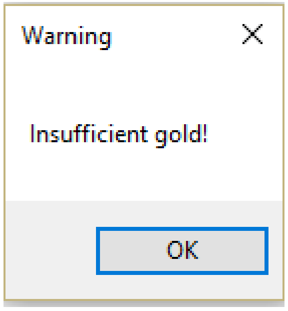
Name: Greatsword
Weight: 15
Value: 100 gold
Base Damage: 2d6

Figure 3: Selecting a Battleaxe and clicking "Buy" results in the Battleaxe removed from shop and added to the player.



Are you sure you wish to exit?

Figure 4: The exit confirmation window. This should be opened when Exit is pressed. If “Yes” is clicked, the application should close, otherwise return to the main form.



Insufficient gold!

Figure 4: Insufficient gold! MessageBox.

You will have 90 minutes to complete your work.

Your name and/or school name should NOT appear on any work you submit for grading.

Submit a copy your entire solution/project to the flash drive provided. You must submit your entire solution/project so that the graders may open your project to review the source code and/or build and execute your solution/project. **Submissions that do not contain source code will not be graded**.

Development Standards

- Standard name prefixes must be utilized for variables.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive	____ 10 pts
The project is named according to the naming conventions	____ 10 pts

Program Execution

Code copied to USB drive and program runs from USB	____ 20 pts

If the program does not execute, then the remaining items in the program executive section receive a score of zero.

The program contains an Exit button that asks for user confirmation	____ 10 pts
The main form does not have handle controls (minimize, maximize, close)	____ 10 pts
On startup, player has four items and 200-400 gold	____ 25 pts
On startup, shop has ten items and 1000-1200 gold	____ 25 pts
Clicking the Refresh Shop button randomly selects ten new items and a new amount of gold between 1000-1200 gold	____ 20 pts
Selecting an item from the player list shows the image and description	____ 20 pts
Selecting an item from the shop list shows the image and description	____ 20 pts
Clicking the Buy button:	
Player gold is correctly checked	____ 20 pts
If sufficient gold, selected item is transferred from shop to player	____ 20 pts
If sufficient gold, player and shop gold are updated correctly	____ 20 pts
Clicking the Sell button:	
Shop gold is correctly checked	____ 20 pts
If sufficient gold, selected item is transferred from player to shop	____ 20 pts
If sufficient gold, player and shop gold are updated correctly	____ 20 pts
Clicking the Load button opens dialog loads selected player file	____ 25 pts
Clicking the Save button opens dialog saves active player to file	____ 25 pts
Clicking the Auto-Quest button adds 20-40 gold to the player	____ 10 pts

Source Code Review

Code is commented at the top, for each function, and as needed	____ 10 pts
Code uses reasonable and consistent variable naming conventions	____ 10 pts
An “Item” class is implemented with appropriate members	____ 10 pts
A “Weapon” class and an “Armor” class are implemented that inherit from “Item” with override for GetInformationString	____ 10 pts
A “Character” class is implemented with appropriate members	____ 10 pts
Serialization file is imported into project	____ 10 pts
Source to create random character is present	____ 10 pts
Source to generate random shop items is present	____ 10 pts
Source to update Player GUI is present	____ 20 pts
Source to update Shop GUI is present	____ 10 pts
Source to control user interaction (shop/player selection) is present	____ 20 pts

Source to control user interaction (buying/selling items) is present	____ 20 pts
Source to generate random gold value is present	____ 10 pts
Source to load player information (from xml) is present	____ 20 pts
Source to save player information (to xml) is present	____ 20 pts

Total Points: 550 pts