Serialize your Bank Account form the BankUI that we did

Add a button to Save the current account to file and Load it.

The Save button should bring up a SaveFileDialog and the Load button should bring up an OpenFileDialog

Using the file selected save the current account. I used an XmlSerializer here are a few rules that you need to know that we didn’t cover in class

1. All the classes that are being serialized need to be public
2. The class needs to have a default constructor, including all the classes in it (the Transaction, Account, Savings and Checking class needs to be public)
3. The fields need to be properties and they need to have a get and set

The Transaction, Account, Savings and Checking class needs to be public have a default constructor and their properties need a get and set, for them to be serialized

If we are using derived classes like our Account and Checking, Savings classes and you are serializing the Account which is really a Checking account, you need the Account to have the attribute of [XmlInclude(typeof(Checking)), XmlInclude(typeof(Savings))] on the Account class definition or when creating the XmlSerializer for the Account create it this way :

XmlSerializer formatter = new XmlSerializer(typeof(Account), new Type[] { typeof(Checking), typeof(Savings)});

formatter.Serialize(fileStream, this); //this being the account

Or you can just serialize the Account’s transactions (not the whole Account) this way:

XmlRootAttribute root = new XmlRootAttribute("Results");

XmlSerializer formatter = new XmlSerializer(typeof(Transaction[]), root );

formatter.Serialize(fileStream, Transactions);

and then deserialize:

Transaction[] trans = (Transaction[])formatter.Deserialize(fs);