Assignment: MySavings
Course: CSE2130 - File Structures and Exception Handling

How has your program changed from planning to coding to now? Please explain?

Originally I had tried to use code from the Roster skillbuilder to complete the assignment, the only thing I had trouble with was loading the object from a file. Copying the object to the file was straightforward.

```
public static void main(String[] args)
{

    int userChoice = 0, amt;
    Scanner input = new Scanner(System.in);

    //PiggyBank account1 = new PiggyBank();

    PiggyBank account = load();
```

As seen in the picture above, the comments represent my previous code/error, I had created a new PiggyBank object each time the code was ran which meant it reset the data and changed the file each time it ran. To fix this I created methods using object/file output/input stream to save and load the object. This was something I wasn't expecting to do and definitely didn't plan to do.

```java
    //Save and load to file methods for PiggyBank object

    // Method to save object to file with output stream
    private static void save(PiggyBank account)
    {
        try (ObjectOutputStream writeAcc = new ObjectOutputStream(new FileOutputStream(bankFile)))
        {
            writeAcc.writeObject(account);
            System.out.println("PiggyBank saved successfully");
        }
        catch (IOException e)
        {
            System.out.println("File could not be created. Error saving PiggyBank");
            System.err.println("IOException: " + e.getMessage());
        }
    }

    // Method to load object from a file with input stream
    private static PiggyBank load()
    {
        try (ObjectInputStream readAcc = new ObjectInputStream(new FileInputStream(bankFile)))
        {
            return (PiggyBank)readAcc.readObject();
        }
        catch (IOException | ClassNotFoundException e)
        {
            // If the file doesn't exist, return new PiggyBank object
            System.out.println("No saved PiggyBank. Loading a new one.");
            return new PiggyBank();
        }
    }

}
```

My biggest problem was trying to load the object each time the code was ran but this was solved by using methods.