Gregorio Loi
September 26, 2021
CS 3331 – Advanced Object-Oriented Programming – Fall 2021
Daniel Mejia
Programming Assignment #2

I confirm that the work of this assignment is completely my own. By turning in this assignment, I declare that I did not receive unauthorized assistance. Moreover, all deliverables including, but not limited to the source code, lab report and output files were written and produced by me alone.

1. Program Explanation

In this section, explain the overview of the assignment.
What did you do?

- For this lab I made a program that handles all the issues presented for the lab, such as taking care of customer accounts, retrieving information for specific events, etc.

How did you tackle the problem?

- To tackle the problem, I took it one step at a time. First, I laid out the infrastructure for the data indexes, as well as handle the customer interactions as I went along.

What techniques did you use to solve the problem?

- I used various techniques along the way, such as using scanner implementations and method work dividing to help conquer and divide.

Did you break the problem into smaller problems? Explain.

- Yes! I didn’t tackle the problem as a whole, I broke it down into several parts. I managed to get rid of each tasklist as I went along chronologically of what was needed to be done.

2. What did I learn?

What did you learn as a result of this assignment?

- I learned some interesting techniques for organization, though I believe this program could’ve been done better.
- My variables could have realistic data types (Several integer values for date and time rather than just being a string), I could make an interface to ask if the user wants any changes made or if the user wants anything new added to the Event class.

What ideas do I have about another way to solve the problem?

- Another idea about another way to solve the problem could be finetuning the customer interaction and allowing more and more interactions.

How long did it take me to complete this lab assignment?

- About one-two days total.

3. Solution Design

What did I do in this program?

- I solved the issue of reading the CSV File using scanner and String.split! To solve the issue of storing each event object for organization, I made an ArrayList of type <Sport> which surprisingly worked. To solve the issue of accumulating logs, I learned about static variables which are different from Instance variables!

What was my approach to solving this problem?

- My approach was very direct and “Divide and Conquer”-esque. I didn’t worry about any issues or bugs until I encountered them, I just focused on little tasks such as creating a Event object to test its attributes, then proceeding to create Event objects in bulk and testing their attributes!

What data structures did I use? Why?

- I primarily used ArrayLists for this implementation because of how easy they are to navigate as well as create.

What assumptions, if any, did I make?

- I assumed that the customer was an average person who needed easy interaction with the system
- I assumed that all file paths were needed to be asked for
- I assumed that main menu suggested the first couple of questions asked

4. Testing

How did I test my program?

- I tested my program through various stages of development, little by little. When I created account indexing, I tested it myself by inputting various account informations from the CSV File

Did I use black-box, white-box testing, or both? Why?

- I used both. The reason I used both is because of the vast amounts of Event objects we had to create (Over 40). So without looking at the data in the various amounts and making sure of each attribute, I simply created Event objects in bulk and tested some for accuracy.

Did I test my solution enough? How can my testing practices be improved?

- Yes! I did rigorous testing throughout the development process.

What are the test cases I used?

- Testing whether Event object attributes represented accurate Event ID associated
- Testing whether account information, indexing, etc. Were all accurate
- Testing whether CSV File Read was accurate and not misplaced
- Testing whether Log files were replicated across instances rather than localized
- Testing whether Log.txt was successfully being written to and perfectly matched the actual log history

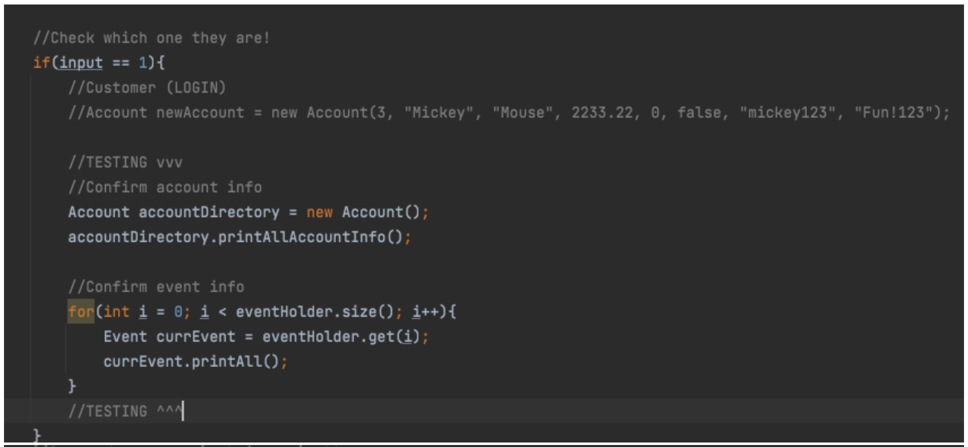
Did I break my program and use that as a way to improve it?

- No, not this time.

5. Test results

Describe the results of your tests.

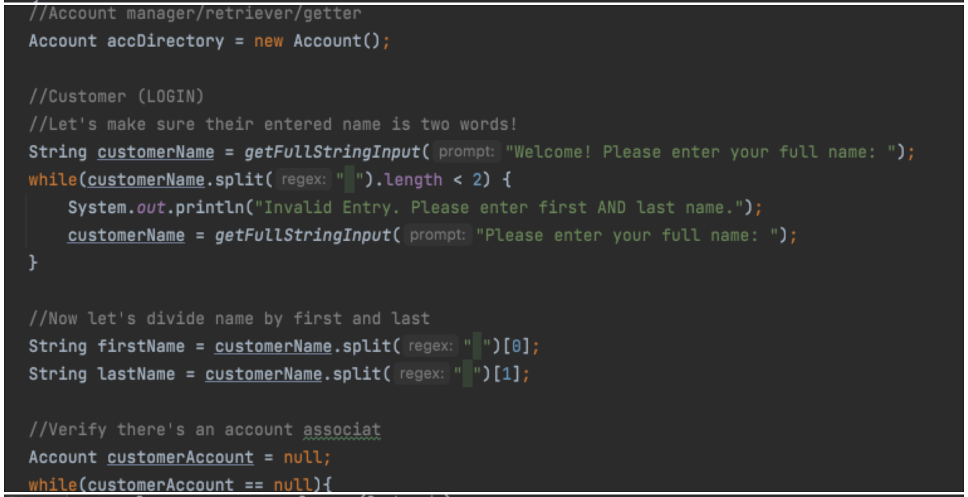
- Besides the massive amount of coding that was required, I didn’t have that many errors.
- I realized that my approach of doing “scnr.close()” wasn’t at all optimistic, it was in fact ruining my print scheme so I had to get rid of it
- I used weird testing methods such as making my own account directory, etc.
- When asking for a customers full name, I had issues ranging from making sure the customer inputted a full name to separating it by first and last



```
//Check which one they are!
if(input == 1){
    //Customer (LOGIN)
    //Account newAccount = new Account(3, "Mickey", "Mouse", 2233.22, 0, false, "mickey123", "Fun!123");

    //TESTING vvv
    //Confirm account info
    Account accountDirectory = new Account();
    accountDirectory.printAllAccountInfo();

    //Confirm event info
    for(int i = 0; i < eventHolder.size(); i++){
        Event currEvent = eventHolder.get(i);
        currEvent.printAll();
    }
    //TESTING ^^^
}
```

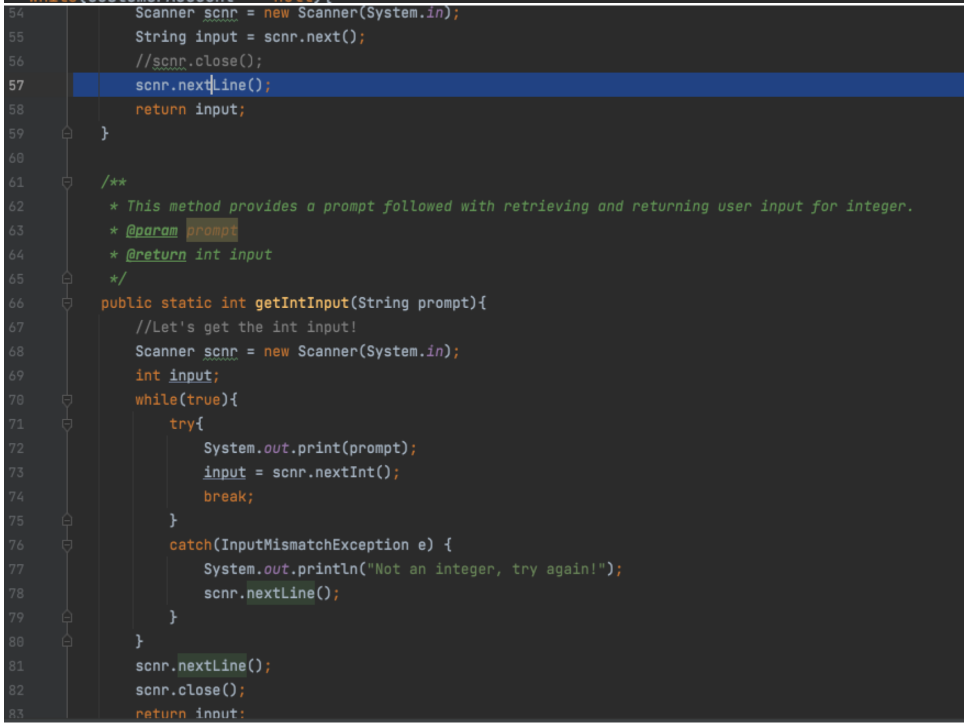


```
//Account manager/retriever/getter
Account accDirectory = new Account();

//Customer (LOGIN)
//Let's make sure their entered name is two words!
String customerName = getFullStringInput( prompt: "Welcome! Please enter your full name: ");
while(customerName.split( regex: " ").length < 2) {
    System.out.println("Invalid Entry. Please enter first AND last name.");
    customerName = getFullStringInput( prompt: "Please enter your full name: ");
}

//Now let's divide name by first and last
String firstName = customerName.split( regex: " ")[0];
String lastName = customerName.split( regex: " ")[1];

//Verify there's an account associat
Account customerAccount = null;
while(customerAccount == null){
```



```
        Scanner scnr = new Scanner(System.in);
        String input = scnr.next();
        //scnr.close();
        scnr.nextLine();
        return input;
    }

    /**
     * This method provides a prompt followed with retrieving and returning user input for integer.
     * @param prompt
     * @return int input
     */
    public static int getIntInput(String prompt){
        //Let's get the int input!
        Scanner scnr = new Scanner(System.in);
        int input;
        while(true){
            try{
                System.out.print(prompt);
                input = scnr.nextInt();
                break;
            }
            catch(InputMismatchException e) {
                System.out.println("Not an integer, try again!");
                scnr.nextLine();
            }
        }
        scnr.nextLine();
        scnr.close();
        return input;
```

Include any text document output as a result of your tests.