CITP 190 – Intro to Programming in JAVA **Student: ____________________**

Project 12 **PROJECT CHECKLIST**

Item	**Possible Points**	**Earned Points**	**Notes**
Proof • For the "list" command shows what will be displayed • For the "add" command shows what will be in the file	**3**		
Screen capture(s)	**2**		
Program design for Project • The `main()` method: ○ displays the command menu as shown ○ verifies that a valid command has been entered (commands are case sensitive) ○ uses the CustomerIO class to display a readable list of the file contents when the "list" command is entered ○ when the "add" command is entered: ▪ asks the user to enter an email address and validates the email address is not an empty string ▪ asks the user to enter a first name and validates the first name is not an empty string ▪ asks the user to enter a last name and validates the last name is not an empty string ▪ creates an object of the Customer class ▪ uses the CustomerIO class to add the customer to the text file ▪ if the add is successful, a success message is displayed that includes the customer's first and last name ▪ if the add is not successful, an error message is displayed that includes the customer's first and last name. ○ displays the message "Bye" and exits the program when the "exit" command is entered • The Customer class is not changed except for adding the comment block at the top of the code. • The CustomerIO class: ○ must handle all the IO for the file ○ must <u>not</u> require the user to do any data entry ○ must have a method that accepts a Customer object as its parameter and saves that Customer object to the file ○ must be able to read and write to the file that is included as one of the starter files for this Project	**5**		

Item	**Possible Points**	**Earned Points**	**Notes**
Design Diagrams: • A correct class diagram is provided for all classes • Design documentation reflects actual logic of code • All methods are documented (one diagram for each method; you may have more than one diagram on a page) • No diagram is larger than one page (8 ½ by 11 inches with ½ inch margins on all sides) • If using flowcharts to diagram the logic: ○ Each flowchart begins and ends with a terminator symbol Note: the main method beginning terminator contains the word `main()`. The main method ending terminator contains the word `return`. Because you do not write the code that calls the main method, you will not have any flowcharts where the beginning terminator contains the word `START` and the ending terminator contains the word `END`. ○ The appropriate symbol is used ○ Only one task per process symbol (the rectangle); each variable declaration should be in its own symbol; show the entire formula for calculations ○ Every symbol (except a terminator) has at least one flowline leading to it and one and only one flowline leading from it. • If using structured pseudocode to diagram the logic: ○ The pseudocode is appropriately indented ○ Each variable declaration is on its own line ○ The entire formula is shown for calculations ○ Selection and iteration blocks have a clear beginning and ending • If using Warnier Diagrams to diagram the logic: ○ Braces are appropriately labeled ○ Each variable declaration is on its own line ○ The entire formula is shown for calculations	**5**		
Following course standards: • Code standards: ○ Code restricted to 80 columns ○ Follows naming conventions for classes, variables, methods, and constants ○ Appropriate comment block at top of program file (may use javadoc conventions) ○ Methods appropriately commented (may use javadoc conventions) ○ Variables have meaningful names ○ Braces align correctly ○ Control statements formatted correctly • All non-code files contain your name, the course code (CITP 190), and the project number at the top of the file. • All design diagrams are in one file. • All files are in standard 8 ½ by 11 inch format with at least ½ inch margins on all sides of the page.	**5**		
Penalties: • "Borrowing" code from the AddressBookIO class (Project 6) without appropriate comments. (This would constitute plagiarism.) • Incorrect calculations • Output is not presented as shown (including spelling and spacing) • Code does not follow the standards • Not all test data was used • Reflects material outside what has been covered in Chapters 1 through 10 and Chapters 13 and 19. • Using any classes not mentioned in the instructions or does not use one of the classes mentioned in the instructions • Using a `continue` statement or misusing a `break` statement	**-20 for any of the items listed**		

Item	Possible Points	Earned Points	Notes
Total	**20**	**0.0**	