

Contestant Number: _____

Time: _____

Rank: _____



JAVA PROGRAMMING

(340)

REGIONAL 2022

Production:

Program 1: Tollway Customer Database _____ (445 points)

TOTAL POINTS _____ (445 points)

Test Time: 90 minutes

GENERAL GUIDELINES:

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

1. Member must hand in this test booklet and all printouts if any. 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 (handwritten, photocopied, or keyed) are allowed in the testing area.
3. Electronic devices will be monitored according to ACT standards.

You will have ninety (90) minutes to complete your work.

Your name and/or school name should *not* appear on work you submit for grading.

1. Create a folder on the flash drive provided using your contestant number as the name of the folder.
2. Copy your entire solution/project into this folder.
3. Submit your entire solution/project so that the graders may open your project to review the source code.
4. Ensure that the files required to run your program are present and will execute on the flash drive provided.

*Note that the flash drive letter may *not* be the same when the program is graded as it was when you created the program.

*It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter.

The graders will *not* compile or alter your source code to correct for this.
Submissions that do *not* contain source code will *not* be graded.

Assumptions to make when taking this assessment:

- There will only be one record created from the supplied data fields in the comments section.
- Users can attempt to enter in erroneous information during an input prompt.
- The program will only exit at a single point which is when they decline to retrieve the single record created.
- All getter and setter methods for the Customer and Address object have been created; the source code for these classes is not accessible.
- If the user does not want to update the records the program will ask if they want to retrieve the records. If they answer “no” the program terminates. If they answer yes then the output (supplied in this document) will have NO reference ID.

Development Standards:

- Your Code must use a consistent variable naming convention.
- All subroutines (if any), functions (if any), and methods (if any) must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).
- If you create a class, then you must use Javadoc comments.

Tollway Database Customer Entry Prototype

You are being charged with creating a prototype software that will allow a data clerk to create a customer record for a tollway company. The initial stage of this prototype will involve you creating a process to allow the data entry clerk to update completed records; the update will only entail one change which will be entering the customers deposit into their account (NOTE: tollway companies require customers to deposit money into an account that will deduct monies from said account anytime they use one of the tollways; you do not need to worry about this transaction activity). The deposit amount will be required to fall within the range of values as stated in the following section; the exact amount you type when testing the program is not important; what is important is that the deposit amount entered by the data clerk is not less than or greater than the range required by the program. The data clerk will NOT enter in commas or \$ symbols when entering deposits: only the number. Furthermore, the data clerk will also be given an option to retrieve the record (remember we are only using one test case at this level) whereupon the record will be retrieved and printed into the console.

The program will be required to format all money entries into US currency that will include the dollar symbol, commas and periods to separate the dollars and cents. For example: \$4,456.55

In addition, the program will also be required to create a randomized transaction reference ID for quality control. The reference ID will only be created when the data entry clerk updates the record. The format for the randomized reference ID had the following requirements:

- 8 total alpha-numeric values
- The first three indexes of the string must be random capital letters, and the letter “O” must be omitted
- The final five indexes of the string must be a random number within the range of 10000 to 99999
- The final reference ID must a single string object from the letter and number combinations.
- For example: LLG80402

Input/Output:

The first output you will see is a console statement letting you know if your customer object and its nested address object was successfully created. After this, the data clerk will be prompted to enter “Yes” or “No” if they want to update the record.

Customer record import successful.

Type in "Yes" if you want update this record:

ENTER: Yes or No:

Inferring that the user enters “Yes” to update the record, the next output will be prompting the user to enter in how much the customer will be depositing (*remember the exact dollar value entered does not matter, the goal is to not allow the data clerk to enter in less or more than the required range*). NOTE: a response of “No” will move the user to the prompt asking to retrieve the record.

All new records require a new deposit. How much will the customer be depositing?

Please enter in a value between \$1.00 to \$9,999.99: (*enter in any value in the range*)

After the deposit is entered in properly, there will be an output similar to the following that includes the formatted currency and randomized reference ID; this will be followed with a prompt asking the user to enter if they want to retrieve the record:

Jose Montana deposited a total of \$456.00.

Reference ID: JAK28949

Do you want to retrieve this record?

ENTER: Yes or No:

After the record has been completed the example below should be the final printed format when the user enters “Yes” to retrieve the record (Remember: the Deposit and Reference ID will have different values than the ones shown below). NOTE: a response of “No” will tell the user “Goodbye” and end the program.

Jose Montana
3445 Rockhill Rd.
Santa Fe, New Mexico 77777
Deposit: \$3,454.00
Car Information: Ford F150
Reference ID: MVL74117

Sample Run Examples (NOTE: these examples have formatting features such as **bold** and *italicized* fonts to bring your attention to key entry actions by the user; your program is not required to display such formatting)

Sample Run #1.a with Update Entry Error (repeats entry request until Yes or No entered)

Customer record import successful.

Type in "Yes" if you want update this record:

ENTER: Yes or No: *sssss*

Type in "Yes" if you want update this record:

ENTER: Yes or No:

Sample Run #1.b with Retrieval Entry Error (repeats entry request until Yes or No entered)

Type in "Yes" if you want update this record:

ENTER: Yes or No: *no*

Do you want to **retrieve** this record?

ENTER: Yes or No: *dgsdg*

Do you want to retrieve this record?

ENTER: Yes or No:

Sample Run #2.a Update ("Yes") with Numerical Entry Error (repeats entry request until double or integer values entered)

Type in "Yes" if you want update this record:

ENTER: Yes or No: *yes*

All new records require a new deposit. How much will the customer be depositing?

Please enter in a value between \$1.00 to \$9,999.99: *hsfhs*

Please enter a correct value.

Please enter in a value between \$1.00 to \$9,999.99:

Sample Run #2.b Update (“Yes”) with Numerical Entry Error (program truncates all digits past hundredth place value)

Type in "Yes" if you want update this record:

ENTER: Yes or No: **yes**

All new records require a new deposit. How much will the customer be depositing?

Please enter in a value between \$1.00 to \$9,999.99: **45.559**

Jose Montana deposited a total of \$45.55.

Reference ID: WVB28039

Do you want to retrieve this record?

ENTER: Yes or No:

Sample Run #2.c Update (“Yes”) with Numerical Entry Error Beyond Parameters (forces required entry)

Type in "Yes" if you want update this record:

ENTER: Yes or No: **yes**

All new records require a new deposit. How much will the customer be depositing?

Please enter in a value between \$1.00 to \$9,999.99: **33333**

Please enter in a value between \$1.00 to \$9,999.99: **.01**

Please enter in a value between \$1.00 to \$9,999.99: **345.55**

Jose Montana deposited a total of \$345.55.

Reference ID: QIT44464

Do you want to retrieve this record?

ENTER: Yes or No

Requirements:

1. You must create an application with the main class called TollwayCustomerDataBase. This class will create the Customer object and its nested attribute which is the Address object. The Customer and Address classes are already created.
2. The TollwayCustomerDataBase will be the driver program that runs the data entry prompts from the user, formats data, and retrieves information from the Customer object.
3. Your contestant number must appear as a comment at the top of the main source code file.
4. If incorrect data is entered, then the program should display an appropriate message and exit.
5. The program will perform the required tasks correctly for however many stocks are indicated by the user.
6. The program will display the outputs as shown above examples above.
7. Use the list of each class behaviors and attributes (after the rubric) to help you

Solution and Project

The project is present on the flash drive	_____	10 points
The projects main class is named TollwayCustomerDataBase	_____	10 points
The class helper method is named setReferenceID(Customer c)	_____	10 points
The class helper method is named getDepositMessage(Customer c)	_____	10 points
The class helper method is named getUserStringInput()	_____	10 points
The class helper method is named getUserNumberInput()	_____	10 points
The class helper method is named consoleRecordCheck(Customer c)	_____	10 points

Program Execution

The program runs from the USB flash drive	_____	15 points
---	-------	-----------

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

The program displays a message declaring successful record import.	_____	10 points
The program prompts and forces user to enter “yes” or “no” if they want to update the record & it is not case sensitive.	_____	10 points
If “no” is entered for update: the program prompts and forces user to enter “yes” or “no” if they want to retrieve the record & it is not case sensitive.	_____	10 points
If “yes” is entered for update: the program prompts and forces user to enter “\$1.00 to \$9,999.99” the deposit amount. All data entry errors are caught.	_____	10 points
Program displays customer name and how much was deposited properly formatted: i.e. <i>Jose Montana deposited a total of \$56.00.</i>	_____	20 points
Transaction reference ID is randomly generated with the first three elements being letters and the remaining five are 0 to 9. Note: the letter “O” must be omitted. i.e. <i>Reference ID: KMU43187</i>	_____	20 points
The program prompts and forces user to enter “yes: or “no” if they want to retrieve the record & it is not case sensitive.	_____	10 points
If “no” is entered for retrieval: the program prompts says “Goodbye” and the terminates.	_____	10 points
If “yes” is entered for retrieval: the program prints the entire record, including the formatted deposit, and the same randomly generated reference ID.	_____	20 points
Output matches required format.	_____	20 points

Source Code Review

The source code is properly commented		
A comment containing the contestant number is present	_____	10 points
Methods and code sections are commented	_____	20 points

Code uses try... catch for exception handling for getUserNumberInput() when entering numbers. All values entered beyond given range are not accepted; and all values entered into the thousandths decimal place are truncated.	_____	30 points
getUserNumberInput() : All values entered beyond given range are not accepted; and all values entered into the thousandths decimal place are truncated.	_____	20 points
main (String args []) : Customer object is correctly constructed from data entry into its given attributes, and also properly passes data into attributes of Address object (attribute in Customer).	_____	10 points
getDepositMessage(Customer c) method retrieves required fields from Customer and also formats the deposit amount to US currency including "\$" and "," plus the cents.	_____	30 points
setDepositCustomerRecord(Customer c) : calls on <i>getUserNumberInput()</i> for data entry and deposits values into customer object; and calls <i>setReferenceID(Customer c)</i> to create reference ID.	_____	20 points
setDepositCustomerRecord(Customer c) : calls on <i>getDepositMessage(c)</i> for String to print out for feedback to the user.	_____	10 points
setReferenceID(Customer c) : randomly generates three capital letters, and omits the latter "O".	_____	20 points
setReferenceID(Customer c) : randomly generates an integer greater than 9,999 and less than 100,000; concatenates with the three letters and returns the value. The reference ID must be in the following format(L: letter & N: number): LLLNNNNNN	_____	20 points
consoleRecordCheck(Customer c) : retrieves all data from the customer object using its getter methods. Places information in proper format	_____	10 points
consoleRecordCheck(Customer c) : formats the deposit retrieved from customer object into US currency including "\$" and "," plus the cents.	_____	20 points

Total Points _____ /445 points

Class Behaviors and Attributes: Help Information

(NOTE: this is the class you will program)

public class TollwayCustomerDataBase Methods:

- void setDepositCustomerRecord(Customer c): Prompts the user to submit a deposit: calls other helper methods
- void setReferenceID(Customer c): Creates and stores the reference ID (random letters and numbers)
- String getDepositMessage(Customer c): Displays the message in console after a successful deposit entry
- String getUserStringInput(): Gets the user input for the yes or no prompts and turns it into LC
- double getUserNumberInput(): Gets the user input for the deposit prompt
- void consoleRecordCheck(Customer c): Prints the final record to the console by using data from the object

(REMEMBER: you do not have access to the source code for the following classes)

class Customer Attributes:

- String last_Name;
- String first_Name;
- String carMake;
- String carModel;
- String carLicensePlate;
- double deposit;
- String reference = "";
- public MailingAddress address;

class Customer Methods:

- public Customer (String ln, String fn, String cMa, String cMo, String lp, Double de): constructor
- public String getLN(): gets last name
- public void setLN(String): sets last name
- public String getFN(): gets first name
- public void setFM(String): sets first name
- public String getMake(): gets make of car
- public void setMake(String): sets make of car
- public String getModel(): gets model of car
- public void setModel(String): sets model of car

class MailingAddress Attributes:

- String street;
- String city;
- String state;

- String zipCode;

class MailingAddress Methods:

- public MailingAddress(): default constructor, fills all attributes with " "
- public MailingAddress(String street, String city, String state, String zipcode): constructor
- public String getAdd(): returns the complete address