KEY

JAVA PROGRAMMING

(340)

REGIONAL 2020

Production Portion:

Program 1: Point of Sale ________________ (315 points)

TOTAL POINTS _______________ ***(315 points)***

Judge/Graders: Please double check and verify all scores and answer keys!

You will have 90 minutes to complete your work.

Your name and/or school name should *not* appear on any work you submit for grading. Write your contestant number in the provided space. Staple all pages in order before you turn in your test.

To submit your project:

1. Create a folder on the flash drive using your contestant number as the name of the folder.
2. Copy your entire solution/project into this folder, including source code.
3. Ensure all the files required to run your program are present on the flash drive and will run on the flash drive.

Use the default package for the Java projects you create.

Point of Sale

You have been hired to create a Point of Sale system for a local coffee shop. The system will allow a cashier to login with a randomly required password and then enter everything a customer is purchasing. When a customer is done, it will provide the total cost after tax. The system will then be ready for the next customer without forcing the cashier to re-enter a password.

General requirements:

- The wording of the inputs and outputs do not need to match the sample run's inputs and outputs exactly. However, make sure they have the same contents and follow the other formatting rules below.
- There should be 1 line of space everywhere there is a line of space in the sample run.
- All prices must have a dollar sign at the front and exactly 2 decimal places.
- All inputs must be checked for errors. Users may enter multiple words when the program is expecting a number, they may enter a number when the program is expecting a letter, or they may enter a decimal when the program is expecting an integer.
- All inputs must be entered on the same line as the prompts, and all prompts must end with a space.
- You must use JavaDoc for all methods in your program. It must cover what the method does and the inputs and outputs if there are any. JavaDoc is not required on the class or variables.
- Your contestant number must appear in a comment near the top of your main class file, somewhere before the start of the main class.
- For yes/no questions, only the first character of the answer matters. Case does not matter. "No!", "n", "N", and "necks are below our heads" all count as "no" to the program because they all start with 'n' or 'N'.

The program will follow the sequence shown in the sample run, though the exact selections and calculations will vary for each customer and run. The overall flow of the program is also given below.

The main class of the program will be called PointOfSale.

Password requirements:

- Before any password checking can be done, a random integer between 10 inclusive and 20 exclusive must be generated. This is the password value.
- The user must enter a string, such as "4 abc 5 43", whose digits add up to the randomly generated number. In the given example, 4+5+4+3=16. This would be a valid password if the randomly generated number was 16. "97", "9 7abc", and "28hi6" would also be valid passwords.
- You must use try-catch in the password checking code.

Menu option actions:

- For menu options 1 and 2, the program must call a separate method that takes in the name of the item ("Regular Coffee" or "Breakfast Croissant"), the starting price of the item, the name of the first option, the price of the first option, the name of the second option, and the price of the second option. This method asks the user which options they'd like and then returns the final price of the item based on the selected options. It must also handle printing the other required lines of text related to the menu item, such as “Regular Coffee:” at the beginning and “Adding Regular Coffee for $8.09.” at the end. A suggested but not required header for this method is “static double getOptions(String itemName, double itemPrice, String opt1Name, double opt1Price, String opt2Name, double opt2Price)”.

- For menu option 9 (total), the program must call a separate method that has the header “static void showTotal(double subtotal)”. The 4 lines of text regarding the subtotal, tax, and total will go in this method. You may also place any spacing lines you’d like to place in this method so that the output looks like the sample run’s output.
- For menu option 0 (exit), the program does not need to go into a separate method.

Program Flow

1. The program begins by requiring the cashier to enter a password whose digits sum up to a random number given in parentheses. This random number must be between 10 inclusive and 20 exclusive.
2. The program next displays the menu. Your menu must match the menu shown in the sample run. The program then asks for the user’s desired menu item.
3. After a menu item is chosen, the program goes into a separate method to get the options for that menu item. The method then returns the final price of the menu item based on the chosen options.
4. The program then displays the menu again and repeats steps 2 and 3 as long as the customer wants additional items.
5. When the user selects option 9 to total the order, the program prints out the subtotal, the tax, and the total after tax. It does not need to print out the individual items the customer is purchasing. This printing also does not need to be in a separate method.
6. The program then returns to step 2, ready for the next customer’s order.

Notes to Graders

- As the password is randomly generated, the number in parentheses will vary. However, it should be between 10 inclusive and 20 exclusive. This can be verified in the submitted code.
- To go along with the random password, you’ll have to enter a password that works with the number in parentheses. The sum of your digits must equal the password number. For example, if the number is 16, you can enter “4 abc 5 4 3”, “97”, “28hi6”, or anything else whose digits add up to 16. To test what the code does if you enter the wrong password, type a password whose digits don’t add up to the number in parentheses.

Test Case

```
Enter password (12): 543 2hi
Error: Incorrect password
Enter password (12): 8me4

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
```

```
Subtotal: $0.00
Your selection: Hi!
Error: Invalid input
Your selection: 3
Error: Invalid input
Your selection: 1

Regular Coffee
Do you want cream for $0.10 more (y/n)? maybe
Error: Invalid input
Do you want cream for $0.10 more (y/n)? y
Do you want sugar for $0.15 more (y/n)? Y
Adding Regular Coffee for $8.24.

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
Subtotal: $8.24
Your selection: 2

Breakfast Croissant
Do you want ham for $0.50 more (y/n)? Y
Do you want cheese for $0.25 more (y/n)? yes I do
Adding Breakfast Croissant for $6.74.

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
Subtotal: $14.98
Your selection: 2

Breakfast Croissant
Do you want ham for $0.50 more (y/n)? nope
Do you want cheese for $0.25 more (y/n)? nah
Adding Breakfast Croissant for $5.99.

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
Subtotal: $20.97
```

```
Your selection: 9

Subtotal:       $20.97
Tax (10%):      $2.10
Total:          $23.07
Please have the customer pay $23.07. Thank you!

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
Subtotal: $0.00
Your selection: 1

Regular Coffee
Do you want cream for $0.10 more (y/n)? n
Do you want sugar for $0.15 more (y/n)? n
Adding Regular Coffee for $7.99.

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
Subtotal: $7.99
Your selection: 9

Subtotal:       $7.99
Tax (10%):      $0.80
Total:          $8.79
Please have the customer pay $8.79. Thank you!

Menu:
1) Regular Coffee ($7.99)
2) Breakfast Croissant ($5.99)
9) Total the order
0) Exit
Subtotal: $0.00
Your selection: 0

Goodbye!
```

Solution and Project

The project is present on the flash drive. _____ 10 points

The project's main class is called PointOfSale. _____ 10 points

Program Execution

The program runs from the USB flash drive. _____ 10 points

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

The program randomly generates a password number between [10, 20) at the beginning. _____ 10 points

The password section rejects strings with sums that don't match the randomly generated number by displaying "Error: Incorrect password", ignores non-numeric characters in the calculation, and uses try-catch. _____ 30 points

The program displays the menu. _____ 10 points

The program asks for cream ($0.10) and sugar ($0.15) for a Regular Coffee and applies the correct price changes based on the answer. _____ 20 points

The program asks for ham ($0.50) and cheese ($0.25) for a Breakfast Croissant and applies the correct price changes based on the answer. _____ 20 points

All inputs are entered on the same line as the prompts. _____ 20 points

There is a 1-line space before getting the options for a menu item. _____ 5 points

There is a 1-line space after getting the options for a menu item. _____ 5 points

There is a 1-line space before displaying the tax and total info. _____ 5 points

There is a 1-line space after displaying the tax and total info. _____ 5 points

There is a 1-line space before displaying "Goodbye!" for menu option 0. _____ 5 points

The program correctly calculates the subtotal, tax, and total. _____ 10 points

The program has 1 or more tab characters in each line of the subtotal, tax, and total lines of showTotal()'s output. It's fine if the tabs don't line up. _____ 10 points

The program displays all prices with a starting '$' and exactly 2 decimal places. _____ 10 points

The program displays "Error: Invalid input" anytime there is an invalid input that's not related to the password. _____ 20 points

Source Code Review

The contestant number is in a comment above the start of the main class. _____ 10 points

A single method that takes in the item name and starting price, option 1 name and price, and option 2 name and price handles displaying info about adding a menu item to the bill and getting the options for the item. _____ 30 points

A method with the header “void showTotal(double subtotal)” handles displaying the 4 lines of text relating to the subtotal, tax, and total for menu option 9. _____ 30 points

JavaDoc is used to cover what each method does and the inputs and outputs if there are any. Not required on class and variables. _____ 30 points

Total Points = _____ **/ 315 points**

<u>Suggested Solution</u>

```
import java.text.DecimalFormat;
import java.util.Random;
import java.util.Scanner;

// Contestant Number = contestant number
public class PointOfSale
{
   // Variables used in multiple methods
   static DecimalFormat moneyFormat;
   static Scanner sc;

   /**
    * Runs when the program starts.
    *
    * @param args Input arguments when running the program. Not used.
    */
   public static void main(String args[])
   {
      moneyFormat = new DecimalFormat("$0.00");
      sc = new Scanner(System.in);

      // Generate password
      Random rand = new Random();
      int pw = rand.nextInt(10) + 10; // [10, 20)

      // Have user enter password
      boolean goodPW = false;
      while (!goodPW)
      {
         int total = 0;
         System.out.print("Enter password (" + pw + "): ");
         String input = sc.nextLine().trim();
         for (int i = 0; i < input.length(); i++) // for each input char
         {
            try
            {
               int value = Integer.parseInt(input.substring(i, i + 1));
               total += value;
            }
            catch (NumberFormatException ex)
            {
            }
         }
         if (total == pw)
            goodPW = true;
         else
            System.out.println("Error: Incorrect password");
      }
      System.out.println();

      // Run the main part of PointOfSale
      while (true) // for each customer
      {
         double subtotal = 0;
         boolean finishedWithCustomer = false;
         while (!finishedWithCustomer) // while the customer isn't yet finished
         {
            // Display the menu
            System.out.println("Menu:");
            System.out.println("1) Regular Coffee ($7.99)");
            System.out.println("2) Breakfast Croissant ($5.99)");
            System.out.println("9) Total the order");
            System.out.println("0) Exit");
            System.out.println("Subtotal: " + moneyFormat.format(subtotal));

            String input;
            boolean goodInput = false;
            while (!goodInput) // while the user hasn't typed a valid menu selection
            {
               System.out.print("Your selection: ");
               input = sc.nextLine().trim();
               switch (input)
               {
                  case "1":
                     subtotal += getOptions("Regular Coffee", 7.99, "cream", 0.1, "sugar", 0.15);
                     goodInput = true;
```

```
                     break;
                  case "2":
                     subtotal += getOptions("Breakfast Croissant", 5.99, "ham", 0.5, "cheese", 0.25);
                     goodInput = true;
                     break;
                  case "9":
                     showTotal(subtotal);
                     goodInput = true;
                     finishedWithCustomer = true;
                     break;
                  case "0":
                     System.out.println("\nGoodbye!");
                     System.exit(0);
                  default:
                     System.out.println("Error: Invalid input");
               }
            }
         }
      }
   }

   /**
    * Offers the user 2 options and calculates the total item price based on the selected options.
    *
    * @param itemName The name of the item.
    * @param itemPrice The starting price of the item.
    * @param opt1Name The first option.
    * @param opt1Price The price of the first option.
    * @param opt2Name The second option.
    * @param opt2Price The price of the second option.
    * @return The total price of the item based on the selected options.
    */
   public static double getOptions(String itemName, double itemPrice, String opt1Name, double
opt1Price, String opt2Name, double opt2Price)
   {
      System.out.println();
      System.out.println(itemName);

      boolean goodInput = false;
      while (!goodInput) // while we don't have good input for option 1
      {
         System.out.print("Do you want " + opt1Name + " for " + moneyFormat.format(opt1Price) +
" more (y/n)? ");
         String input = sc.nextLine().trim().toLowerCase();
         if (input.startsWith("y"))
         {
            itemPrice += opt1Price;
            goodInput = true;
         }
         else if (input.startsWith("n"))
         {
            goodInput = true;
         }
         else
         {
            System.out.println("Error: Invalid input");
         }
      }

      goodInput = false;
      while (!goodInput) // while we don't have good input for option 2
      {
         System.out.print("Do you want " + opt2Name + " for " + moneyFormat.format(opt2Price) +
" more (y/n)? ");
         String input = sc.nextLine().trim().toLowerCase();
         if (input.startsWith("y"))
         {
            itemPrice += opt2Price;
            goodInput = true;
         }
         else if (input.startsWith("n"))
         {
            goodInput = true;
         }
         else
         {
            System.out.println("Error: Invalid input");
         }
      }

      System.out.println("Adding " + itemName + " for " + moneyFormat.format(itemPrice) + ".\n");
      return itemPrice;
   }

   /**
    * Shows the subtotal, tax, and total to the user.
    *
    * @param subtotal The subtotal to start with in the calculations.
    */
   public static void showTotal(double subtotal)
   {
      System.out.println();
```

```
        System.out.println("Subtotal:\t" + moneyFormat.format(subtotal));
        System.out.println("Tax (10%):\t" + moneyFormat.format(subtotal * 0.1));
        System.out.println("Total:\t\t" + moneyFormat.format(subtotal * 1.1));
        System.out.println("Please have the customer pay " + moneyFormat.format(subtotal * 1.1) + ". Thank you!");

        System.out.println();
    }
}
```