

IS 2033 - BBCA1 ASSIGNMENT

TOTAL POINTS: 100
CODE DUE: 2/29/2020 BY 11:59 PM Do not submit code that doesn't compile, run, or generate correct output.
PLAN DUE: TO BE GRADED BY EMAILING YOUR PLAN TO ONE OF THE ACADEMIC COACHES NO LATER THAN DAY BEFORE CODE DUE DATE, 2/28/2020. DO NOT WAIT: OTHERWISE, YOU WON'T GET A GRADE!!! ALL OR NOTHING. 10% OF GRADE.
READ CAREFULLY *ALL* THE INSTRUCTIONS!!!

UTSA HONOR CODE: As a UTSA student, you are **bound** by the honor code, so DO NOT cheat on any of your coursework. **By submitting this assignment, you are attesting to your own authorship** based on material from the IS 2033 textbook and/or your professor. Cheating can result in any one, or combination, or all of the following: reduced or failing grade for the assignment, a signed statement of the infraction, reduced or failing grade for the course, reporting of student name to the Department Chair and faculty, Dean's Office and COB faculty, and/or elevation to Student Conduct and Community Standards.

OBJECTIVE (*this is not the program purpose*): Code a program that uses the concepts covered in chapters 1-3 and lecture.

PREP WORK: 1, 2, 3 (includes Shepherd's PowerPoint slides); all labs up through chapter 3.

GRADING: You'll be graded on *how well* you follow the program instructions and the **accuracy of your output as reflected in the prompts, the output specifications, and the sample output. This includes spacing and line advancing. Each line of output can be associated with multiple points in the code!** The instructions, prompts and output are what the user wants. You are **not** at liberty to change anything, but code to these requirements. You'll also be graded on the code given to you by your professor for this BBCA.

PROGRAM INSTRUCTIONS: Create a program that allows a user to purchase one or more of the candies listed.

1. Logical Control Structures:

- Use a **while** loop to process for multiple candy purchases. Use a sentinel-controlled loop variable.
- Use **if-else**, **nested if-else** and **if** structures to figure out
 - whether the candy choice is within the proper range of 1 through 5;
 - the name of the candy;
 - the price of the candy;
 - the formatting of the cost of the first candy item on the sales receipt with a \$ sign;
 - when to print the error message **"Invalid candy choice! Try again."**
- The customer can purchase multiple candies on the same sales receipt.
- The sales receipt is created in real time.
 - Each candy purchase with quantity, candy name, price and item cost is added.
 - The subtotal is calculated with each purchase.
- Once there are no more purchases, the sales tax is calculated along with the total and added to the sales receipt which is finalized for printing/displaying.

- f. Use printf() with format specifiers where needed.
- g. Don't forget to insert the exit statement at the end of main().
- h. The [prompts](#), the [final output specs](#), and the [sample output](#) show you in what order to place your code. To return from these links press Alt then left arrow.

2. **CODE FOR PROPER OUTPUT ALIGNMENT:** where *salesReceipt* is a String variable that holds the current purchases. Note in the code below the use of right and left justification to align the output regardless of the candy and quantity chosen. Note the use of += which reassigns salesReceipt to the new String object created by String.format(). ALL Java statements are to be typed **exactly** as shown. The following **if-else** prints a \$ sign for the first line item; otherwise, there's no \$ sign printed. Look at the output specs above or the sample output. The **trigger** variable determines the first line item. NOTE: There is a **space** between the double quotes.

*/*Java code for [Prompt 2](#)*/*

```
System.out.printf("%nFAIRYTALE SWEETS"
    + "%n%n1.  Arabian Nights Chocolate Coins - 1 lb. Bag %5s%,7.2f"
    + "%n2.  Beauty and the Beast Lollipops - 1 lb. Bag %,12.2f"
    + "%n3.  Mad Hatter Jelly Beans - 1 lb. Bag %,20.2f"
    + "%n4.  Pinocchio's Candy Cones - Each %,23.2f"
    + "%n5.  Sleeping Beauty Caramel Apples - Each %,17.2f"
    + "%n%nEnter your choice: ", "$", 2.25, 2.50, 1.75, 0.75, 1.25);
```

*/*After the purchases are successfully completed, this if-else is used to add the first purchase to the sales receipt with a \$ sign when the formatFirstItem is 1 and no \$ sign when the formatFirstItem is not. This is located in the if body of the if-else for choice < 6. ALL Java statements are to be typed exactly as shown. The following if-else prints a \$ sign for the first line item; otherwise, there's no \$ sign printed. Look at the output specs above or the sample output. The trigger variable determines the first line item. NOTE: There is a space between the double quotes.*

```
*/
if(formatFirstItem == 1)
{
    salesReceipt += String.format("%n%s"
        + "%n      %d @ $%.2f ea. %-24s $%,10.2f%n", candy,
            quantity, price, " ", itemTotal);

    } formatFirstItem = 0;
else
{
    salesReceipt += String.format("%s"
        + "%n      %d @ $%.2f ea. %-25s $%,10.2f%n", candy, quantity,
            price, " ", itemTotal);

} //END if formatFirstItem is 1 OR else formatFirstItem NOT 1
/*This is located outside of the while loop.
*/
salesReceipt += String.format("%n%36s %-6s $%,10.2f"
    + "%n%36s %-7s %,10.2f"
    + "%n%n%36s %-6s $%,10.2f%n", "SUBTOTAL:  ", " ",
        subtotal, "TAX @ 8.125%:  ", " ", taxAmount,
```

```
"TOTAL: ", " ", total);
```

```
System.out.printf("%s", salesReceipt);
```

3. **Develop (pseudocode) your plan FIRST.** The plan is worth 10% of your grade.
 - a. The prompts tell you what input variables you will need.
 - b. The output will tell you the type of calculations you will need (if any) and whether you will need to declare additional variables.
 - c. The output will tell you the order of logic for your code.
 - d. Save your plan as *YourLastNameFirstInitialSecNoBBCA1Plan*. Download the plan instructions and the plan template through the links posted for this assignment on Blackboard.
4. **Work and submit this program on your own** (no partner). Name your program as ***YourLastNameFirstInitialYourSectionNoBBCA1***.
5. **Commenting Your Program:**
 - a. In your program, YOU MUST insert a **program purpose** in the first comment box. The content of that first comment box was shown to you in the *Anatomy of a Java Program* lecture for chapter 1.
 - b. Use Javadoc comment boxes beginning with `/**` and ending with `*/` for your comment boxes.
 - c. Insert a Javadoc comment box above your methods explaining what is going on in the method that goes for the `main()` which is a method.
 - d. Line comment the import statements and the variables declared at the class level and/or in any method [including `main()`].
6. **Formatting Rules:** Refer to the *Java Style Guide* PDF posted on [Blackboard](#) in IS 2033. Always test your output to validate that your program is functioning properly with the correct output and spacing.

PROMPTS: Code the **bold** from the prompts below in the `printf` statements that capture data into your program. Once again, the prompts tell you your input variables. Except for the first prompt, all the other ones reside within the while loop.

1st Prompt: The value captured from this prompt is the loop-control variable for the sentinel-while loop mentioned in 1a of the Program Instructions section above.

Do you want to proceed with your candy purchase? Enter 'Y' or 'N':

2nd Prompt: Displays when the answer to the 1st prompt is 'Y'; otherwise, program ends and sales receipt is printed. Link to prompt in actual output.

FAIRYTALE SWEETS

1. Arabian Nights Chocolate Coins - 1 lb. Bag	\$ 2.25
2. Beauty and the Beast Lollipops - 1 lb. Bag	2.50
3. Mad Hatter Jelly Beans - 1 lb. Bag	1.75
4. Pinocchio's Candy Cones - Each	0.75
5. Sleeping Beauty Caramel Apples – Each	1.25

Enter your choice:

If the choice is 0 or less or greater than 5, display this message:

Invalid candy choice! Try again.

3rd Prompt: If candy choice is 1 through 5, then assign the name of the candy to its variable and the candy's price to a price variable and prompt for the quantity. The Xs is the name of the candy.

Quantity for XXXXXXXXXXXXXXXX:

4th Prompt: The value for this prompt is captured in the loop-control variable for prompt 1.

Would you like to make another candy purchase? Enter 'Y' or 'N':

Final Output Specifications: The Xs is the name of the candy, the ZZ9 is the quantity, the \$Z.99 is the price of the candy and the \$Z,ZZ9.99 is the line item total for that candy purchase. Z's denote printing of 1-9 in that position (suppression of leading 0's). 9's denote printing 0-9 in that position. The XX for Time indicates AM or PM. Use `System.out.printf()` and the appropriate format specifiers (look in Appendix I) to properly space the output. **NOTE:** The date is NOT to be hard coded into the header, instead, you will need to capture the system's date (Appendix I). This is so the date will change accordingly. Furthermore, create your receipt as a String which allows you to add the header and date information first, then add the purchase as the customer selects it. Note how the \$ sign only prints for the first line item on the receipt. Make sure the content of your receipt is properly aligned. Use left and right justification.

FAIRYTALE SWEETS
North Star Mall San
Antonio, TX

← 3 Header Lines: First line of header triple-lined advance (2 blank line before printed line).

Date: 99/99/99
Time: 99:99:99 XX

← Date & Time Labels

XXXXXXXXXXXXXXXXXXXXX
ZZ9 @ \$Z.99 ea.
XXXXXXXXXXXXXXXXXXXXX
ZZ9 @ \$Z.99 ea.

\$Z,ZZ9.99

← Line Items: candy name, quantity, price,
line item total.

Z,ZZ9.99

SUBTOTAL: \$Z,ZZ9.99
TAX @ 8.250%: ZZ9.99

← Subtotal: Sum of line item totals.
Tax Amount: .0825 of subtotal
Total: Sum of subtotal and tax amount

TOTAL: \$ZZ,ZZ9.99

END OF OUTPUT SPECIFICATIONS

SUBMISSIONS REQUIREMENT:

1. **Word Document:** Copy your .java code into a Word document and save it with the same name as your program. Upload the document to Blackboard.

2. **Zippping Folders:** Your Java files **must be** in a folder.
 - a. Create a folder named for the program (excluding the file extension).
 - b. Put your **.java**, **.class**, **.java~** files in the folder.
 - c. To zip the folder, point to it then right click and
 - i. **Filzip** if you have it **OR**
 - ii. Click **Send To** then click **Compressed (zipped) Folder**
 - d. Upload your zipped folder to Blackboard.
3. **Uploading to Blackboard:** **Make sure your browser is properly configured for Blackboard (see syllabus).**
 - a. Your submissions are to be uploaded to Blackboard through **Assignments** only.
 - b. **Upload your files no later than the due date by 11:55 pm**; otherwise, you don't have time to recover from any problems and your assignment may not be accepted by Blackboard.
 - c. Check to make sure your submission is uploaded. Please **do not ask your instructor** to check whether your assignment has been uploaded. You can do this yourself. Or upload during a tutoring session when someone can help you.
 - d. If you submit your assignment before the due date, want to make changes or upload additional files, you can **re-upload** your files. Your last submission is the one graded.

NO ASSIGNMENTS WILL BE ACCEPTED LATE OR VIA E -MAIL. DO NOT UPLOAD PROGRAMS THAT DON'T COMPILE, RUN, or PRODUCE THE CORRECT OUTPUT. Insert the following note in the *Comment* section for the assignment on Blackboard then click Submit: *My program doesn't compile.* OR *My program doesn't run.* OR *My program's output is incorrect.*

*******SAMPLE OUTPUT******* *It is always good to test your code using sample data to see if your program meets the output specifications. **Run your program using the data in the following sample output.** For the output produced by your program to line -up properly, the FONT in DrJava should be Monospaced or Courier New. Copy and paste the output into a traditional comment box `/* */` at the end of your BBKA1.java file. Eliminate the asterisks (*) between the first and the last. The comment box needs to be outside of the close brace for the class. **Worth 5 points!** Your output will not print in bold.*

↓ [Actual Sample Output](#) on next page.

*****OUTPUT WHEN THERE ARE CANDY PURCHASES***** ← NOT PART OF OUTPUT Do

you want to proceed with your candy purchase? Enter 'Y' or 'N': y

FAIRYTALE SWEETS

- | | |
|---|---------|
| 1. Arabian Nights Chocolate Coins - 1 lb. Bag | \$ 2.25 |
| 2. Beauty and the Beast Lollipops - 1 lb. Bag | 2.50 |
| 3. Mad Hatter Jelly Beans - 1 lb. Bag | 1.75 |
| 4. Pinocchio's Candy Cones - Each | 0.75 |
| 5. Sleeping Beauty Caramel Apples - Each | 1.25 |

Enter your choice: 6

Invalid candy choice! Try again.

Would you like to make another candy purchase? Enter 'Y' or 'N': y

FAIRYTALE SWEETS

- | | |
|---|---------|
| 1. Arabian Nights Chocolate Coins - 1 lb. Bag | \$ 2.25 |
| 2. Beauty and the Beast Lollipops - 1 lb. Bag | 2.50 |
| 3. Mad Hatter Jelly Beans - 1 lb. Bag | 1.75 |
| 4. Pinocchio's Candy Cones - Each | 0.75 |
| 5. Sleeping Beauty Caramel Apples - Each | 1.25 |

Enter your choice: 0

Invalid candy choice! Try again.

Would you like to make another candy purchase? Enter 'Y' or 'N': y

FAIRYTALE SWEETS

- | | |
|---|---------|
| 1. Arabian Nights Chocolate Coins - 1 lb. Bag | \$ 2.25 |
| 2. Beauty and the Beast Lollipops - 1 lb. Bag | 2.50 |
| 3. Mad Hatter Jelly Beans - 1 lb. Bag | 1.75 |
| 4. Pinocchio's Candy Cones - Each | 0.75 |
| 5. Sleeping Beauty Caramel Apples - Each | 1.25 |

Enter your choice: 1

Quantity for Arabian Nights Chocolate Coins: 5

Would you like to make another candy purchase? Enter 'Y' or 'N': y

FAIRYTALE SWEETS

- | | |
|---|---------|
| 1. Arabian Nights Chocolate Coins - 1 lb. Bag | \$ 2.25 |
|---|---------|

- | | |
|---|------|
| 2. Beauty and the Beast Lollipops - 1 lb. Bag | 2.50 |
| 3. Mad Hatter Jelly Beans - 1 lb. Bag | 1.75 |
| 4. Pinocchio's Candy Cones - Each | 0.75 |
| 5. Sleeping Beauty Caramel Apples - Each | 1.25 |

Enter your choice: 5

Quantity for Sleeping Beauty Caramel Apples: 2

Would you like to make another candy purchase? Enter 'Y' or 'N':

FAIRYTALE SWEETS

North Star Mall
San Antonio, TX

Date: 02/05/20

Time: 04:12:05 PM

Arabian Nights Chocolate Coins

5 @ \$2.25 ea.	\$	11.25
----------------	----	-------

Sleeping Beauty Caramel Apples

2 @ \$1.25 ea.		2.50
----------------	--	------

SUBTOTAL:	\$	13.75
-----------	----	-------

TAX @ 8.250%:		1.13
---------------	--	------

TOTAL:	\$	14.88
--------	----	-------

*****OUTPUT WHEN THERE ARE "NO" CANDY PURCHASES***** ← NOT PART OF OUTPUT Do

you want to proceed with your candy purchase? Enter 'Y' or 'N': n