Bluegrass Community and Technical College
CIT 149: Java I
Selection Statements

A local artist, Sterling Silver, is selling jewelry at several spring festivals in Kentucky. Items for sell are earrings, necklaces, and bracelets.

> To simply the pricing, the artist is only offering items of equitable quality and price. All items in his booth will sell for $35.00 each.

To generate interest and to increase the traffic at his booth, Sterling is offering a Groupon for his merchandise. Customer can purchase the following Groupons:

- Groupon Code: **Sterling2017**
- Groupon Code: **SILVER2017**

Customers pay for the Groupon <u>in advance</u> and bring the coupon to the spring festival. The **Sterling2017** Groupons cost $20.00 and are good for 1 item at a festival (in other words they paid $20 for a $35 item). The **SILVER2017** Groupons cost $35.00 and allows the customer have 2 items for the price of 1 (in other words they paid $35 for 2 items which would have cost $70). Customers are only allowed to use one Groupon per order.

Write a Java application that Sterling can use to calculate what a customer owes (taxes are included in the prices so no need to add tax).

Because we practiced switch statements last week, this week you will use if structure(s) to solve this problem.

<u>STEP 1</u>:

The program should input

- the full name of the customer (one input which includes first and last name)
- the date the Groupon is purchased (in the format of mm/dd/yyyy)
- the number of jewelry items purchased
- a Groupon code (blank, Sterling2017, or SILVER2017)

If statement can be used to validate user input (error-checking). To that end:

- Make sure the number of items purchased is a positive number greater than zero. If it is not, display an error message letting Sterling know he made a mistake on data entry.
- Make sure the Groupon code is a valid entry (blank, Sterling2017, or SILVER2017). Upper and lowercase matter. If invalid data is entered for a Groupon code, display an error message letting Sterling know he made a mistake on data entry.
- If a **SILVER2017** Groupon code is used, make sure the number of jewelry items purchase is 2 or greater (since the Groupon code assumes they prepaid for 2 items). If that is not the case, display an error message letting Sterling know he made a mistake on data entry.

While you are testing this part, you can let the program end after the error message(s) is displayed. Get this working before continuing with the assignment.

Calculate what the user owes:

- Each jewelry item cost $35. Use a constant for this value.
- If the customer has a Groupon code of **Sterling2017**, subtract 1 from the number of items to be paid for.
- If the customer has a Groupon code of **SILVER2017**, subtract 2 from the number of items to be paid for.
- Calculate what the customer owes.
- Display in a nice format (using the format from the samples below), the name of the customer, date, number of items purchased, any Groupon code they are using, and the total amount owed.

Sample valid output could be (blue for illustration only):

```
Thank you for purchasing jewelry from Sterling Silver, Inc.

Name:    Sally Jo Harrison          Purchase Date:        2/14/2017
Groupon Code: Sterling2017
Number of items purchase:  4
Amount due and paid:  $105.00
```

```
Thank you for purchasing jewelry from Sterling Silver, Inc.

Name:    Billy Bob Johnson          Purchase Date:        2/14/2017
Groupon Code: SILVER2017
Number of items purchase:  3
Amount due and paid:  $35.00
```

```
Thank you for purchasing jewelry from Sterling Silver, Inc.

Name:    Billy Bob Johnson          Purchase Date:        2/14/2017
Groupon Code:
Number of items purchase:  2
Amount due and paid:  $70.00
```

STEP 2:

Once you have this working for 1 customer. **You can add some loops to your program.**

You can use loops to correct data entry errors. Let’s look at a general example. Assume you are asking for number and you only want to accept integers between 1 and 12. The following code would ask for an integer, validate it, and then ask for it again if the value was invalid:

```
Scanner scan = new Scanner(System.in);

int number = -1;              // Note the variable is initialized to an invalid
                              // value so that the loop will execute at least once
                              // and ask for the first value

while (number < 1 || number > 12)
{       System.out.println("Enter a number in the range [1-12] ");
        number = scan.nextInt();
}
```

For the data entry errors you were checking in Step 1, use a loop to allow Sterling to correct an error using an approach similar to that above.

STEP 3:

Now, you want to use a loop to allow Sterling to process multiple customers without having to rerun the program. You can use a **do loop** or **while loop** for this. Look at the format for each below:

Using a while loop – when you do not know if you will have a customer

```
Scanner scan = new Scanner(System.in);
String processACustomer = "yes";

System.out.println("Do you want to process a customer (yes or no)? ");
        processACustomer = scan.nextLine();

while(processACustomer.equals("yes"))
{
        // The code you need to process appears here

        System.out.println("Do you want to process another customer (yes or no)? ");
                processACustomer = scan.nextLine();
}
```

Using a do loop – when you know you have at least one customer

```
Scanner scan = new Scanner(System.in);
String processACustomer = "no";

do
{
        // The code you need to process appears here

        System.out.println("Do you want to process another customer (yes or no)? ");
                processACustomer = scan.nextLine();
}
while (processACustomer.equals("yes"));
```

Add a loop to allow Sterling to process multiple customers.

Step 4:

Before turning in your program, make sure:

- Clean up your code, if needed, and document your program.
- Make sure you used the correct data types and the DecimalFormat method to display the amount due in monetary format. A common mask to use with monetary values is: “$###,##0.00”
- Test your program well with different input values.
- Zip the program before submitting it.