* Create Sales Transaction

1. Console asks user if they would like to create a sales transaction, return item, enter rebate, generate rebate check or close the console. User enters ‘T’ to create sales transaction.
2. The console asks to enter the customer’s *name*. The user enters the customer “Swap Gupta”. The name is saved as a string.
3. The console then asks the user to enter the *item purchased* and the *cost*. The user enters “hammer,10”. The entry is split by the comma and the first part is added to a *list of items* and the second part is added to a *list of costs*. The console then asks the user to enter the next item and enter ‘N’ if there are no more items. The user enters “saw,10”. The entry is split by the comma and the first part is added to a *list of items* and the second part is added to a *list of costs*. The console then asks the user to enter the next item and enter ‘N’ if there are no more items. The user enters ‘N’.
4. The console then asks the user to enter the *transaction ID*. The user enters “1234”. The ID is saved as an int. The console then asks the user to enter the *date*. The user enters “02/16/2018”. The date is saved as a DateTime.
5. The app then will add the transaction to a *dictionary of* ***all transactions***, with the transaction ID’s as keys. The app first checks if the *ID* entered already exists. The *transaction ID* has not been used so the transaction is added to the *dictionary of* ***all transactions***.
6. The app then gets ready to print the sales receipt. First the app builds the transaction string. A new *stringbuilder* is used and appends the *transaction name, ID, and date*. The stringbuilder then appends *all items purchased and their costs*. The transaction totals all costs to make a *subtotal* and that is appended. A *tax* of 10% is appended and then multiplied into the *subtotal* to make the *total*. The *total* is appended. The *stringbuilder* is turned into a string and then the console outputs the transaction string.
7. The output is: “Swap Gupta,1234,02/16/2018, hammer $10, saw $10, subtotal $20, tax 10%, total $22.”

* Return item(s)

1. Console asks user if they would like to create a sales transaction, return item, enter rebate, generate rebate check or close the console. User enters ‘I’ to return item.
2. The console then asks the user to enter the *date*. The user enters “02/16/2018”. The date is saved as a DateTime.
3. The console then asks the user to enter the *transaction ID*. The user enters “1234”. The *ID* is checked to see if the *ID* is saved in the *dictionary of* ***all transactions.*** The *name,* “Swap Gupta”,from the transaction *ID* is saved at a string.
4. The app then starts to return items. The console asks the user to enter the return item. The user enters “hammer”. The app checks if the item is in the transaction *ID,* “1234’. The app then gets the index of the *item* entered in the *list of items.* The index is used to find the associated *cost.* The *item* and the *cost* are saved into *lists of returned items and costs.* The returned *item and cost* are then removed from the transaction. The transaction *total* is then updated to the new total by summing the *costs*.
5. The console asks if the user would like to return another item. The user enters ‘N’.
6. A new transaction object is made using the information from “1234” and the *list of returned items and costs.* The new transaction and the *ID* “1234” are then saved into another *dictionary of returned transactions.*
7. The app then gets ready to print the return receipt. First the app builds the return string. A new *stringbuilder* is used and appends the *transaction name, ID, and date*. The stringbuilder then appends *all items returned and their costs*. The *stringbuilder* is turned into a string and then the console outputs the transaction string.
8. The output is: “Swap Gupta,1234,02/16/2018, Items returned: hammer,$10.”

* Enter rebate

1. Console asks user if they would like to create a sales transaction, return item, enter rebate, generate rebate check or close the console. User enters ‘R’ to enter a rebate.
2. The console then asks the user to enter the *transaction ID*. The user enters “1234”. The *ID* is checked to see if the *ID* is saved in the *dictionary of* ***all transactions.***
3. The console then asks the user to enter the *date*. The user enters “02/16/2018”. The date is saved as a DateTime. *Date* is compared to the date July 15, 2018. *Date* is not after July 15th so the app continues.
4. The *name,* “Swap Gupta”,from the transaction *ID* is saved at a string. The console then asks the user for the *customer address,* the user enters “123 Street, Manhattan,KS”. The console then asks the user for the *customer email,* the user enters “swapgupta@gmail.com”. The console then asks the user for the *discount,* the user enters “20”. *Discount* is divided by 100 to get a percent, 20%. Using the *transaction name, customer address, customer email, date and discount,* a new rebate object is made. The rebate object checks if *date* is in June, but it is not. The new rebate and the transaction are saved in a *dictionary of transactions and rebates.*

* Generate rebate check

1. Console asks user if they would like to create a sales transaction, return item, enter rebate, generate rebate check or close the console. User enters ‘C’ to generate rebate checks.
2. The console then asks the user to enter the *date*. The user enters “12/16/2018”. The date is saved as a DateTime. *Date* is compared to the date July 31, 2018. *Date* is after July 31st so the app continues.
3. Then the app prints rebate check information. First the app builds the rebate check string. The *total* from a transaction is saved as a decimal. The *discount* from a rebate is saved as a decimal. The two decimals are multiplied to get *check amount.* A new *stringbuilder* is used and appends the *transaction name, check amount, and transaction ID*. The stringbuilder then appends *transaction total, the rebate discount, the rebate address and the rebate email*. The *stringbuilder* is turned into a string and then the console outputs the rebate check string.
4. The output is: “Pay to the order of Swap Gupta the amount of $4.40, ID: 1234, total: $22, discount: 20%, address: 123 Street, Manhattan,KS, email: swapgupta@gmail.com.”