

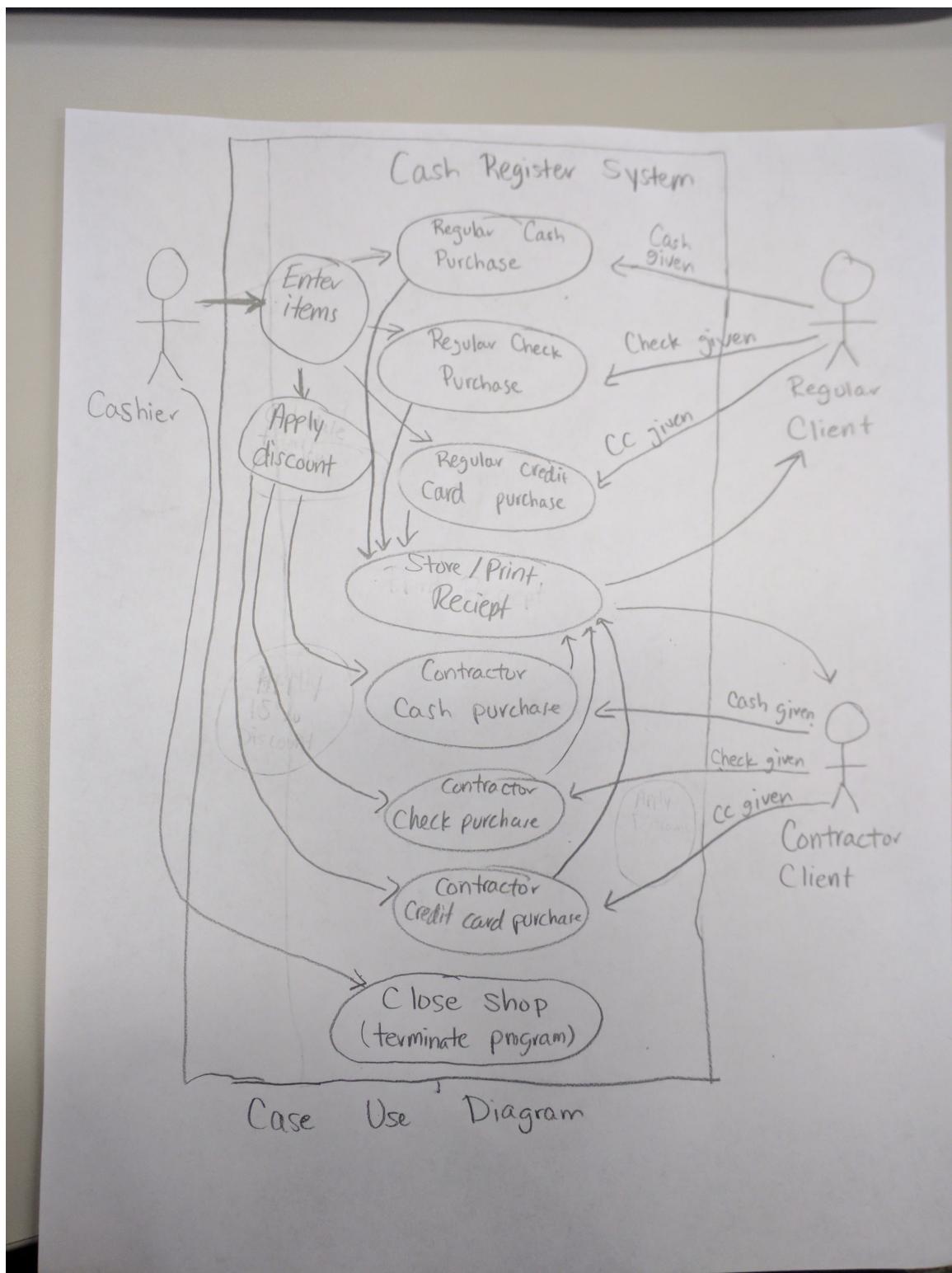
Austin Osborn  
COMP 2710-001  
Lab 4 Design Portion  
[Aeo0015@auburn.edu](mailto:Aeo0015@auburn.edu)

## Analysis (Use Cases)

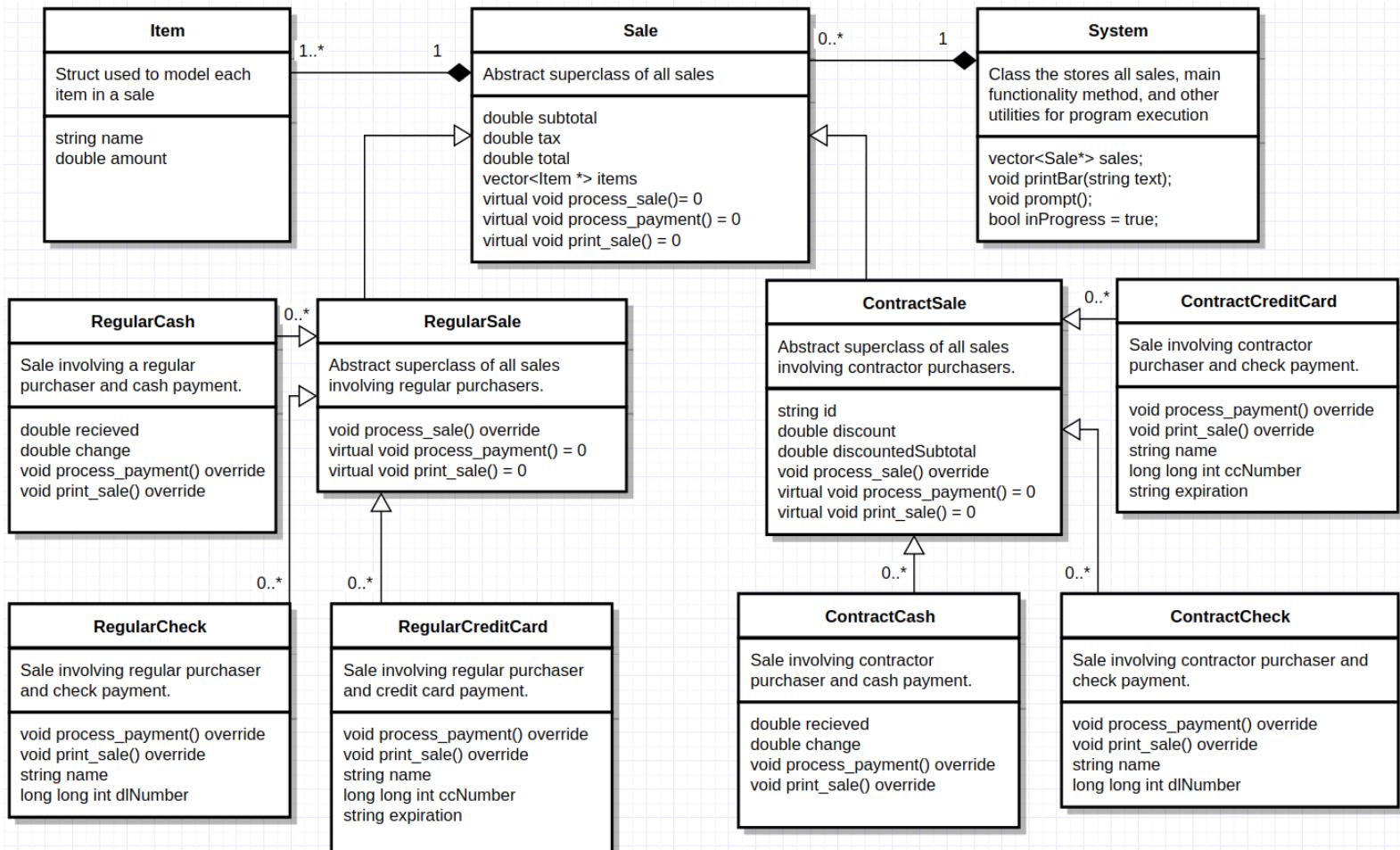
- **Initial execution:** The user is greeted with a selection prompt and they choose appropriate number to select option. This repeats until the user selects option 8 to end the program.
- **Enter a sale:** The cashier wants to enter a sale. They will choose a number (1-6) based on menu UI to make the appropriate type based on how the client wishes to pay. They will be asked to enter items and prices until they enter a \* character, indicating the end of the list. At this point, if the client is a contractor, the cashier is prompted for a contractor ID and a 15% discount is applied to the sale. Tax is calculated. The total is presented. Then, depending on payment method, they enter appropriate information.
  - o Erroneous input will caught throughout and the details of each type can be found in the class diagram and testing section of this document.
- **Print list of sales:** Each sale up to that point in the program is printed out is a list in the order of which they were entered.
- **Quit:** Cashier selects option 8 on the menu and the program ends execution.

## Use Case Diagram

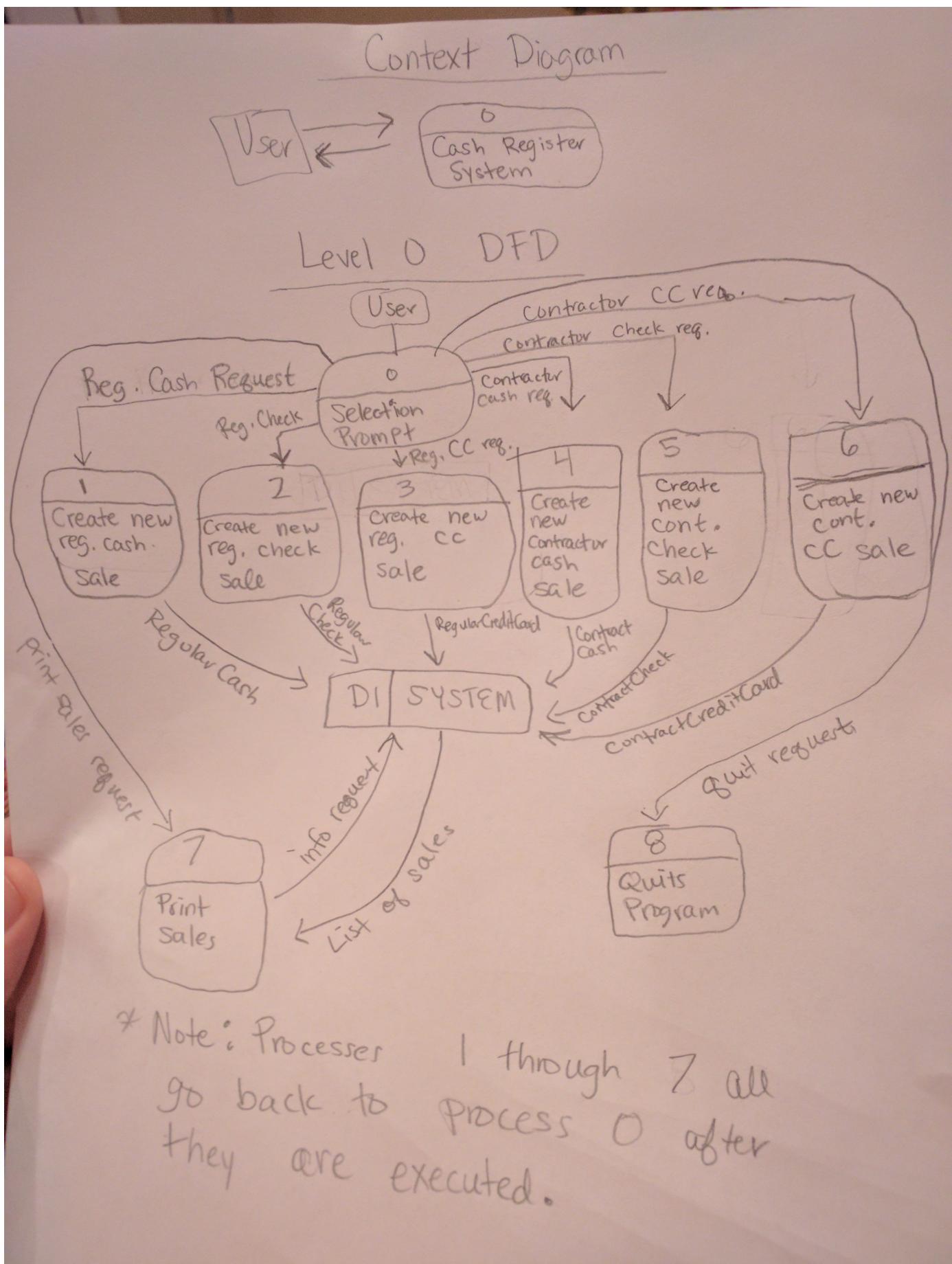
- Below is a use case diagram that might model how the program specified in the lab specifications could be used in a real-world environment.



## Class Diagram



## Data Flow Diagrams



## **Testing (Test Cases)**

### **A. System Wide**

1. Program is first executed.
  - a) User is presented with a menu of options
  - b) User should enter 1 – 8 for their desired selection
    - Any input not conforming to these constraints will initiate a re-prompt.
    - Valid input executes corresponding action.

### **B. Specific Objects/Functions**

1. User selects a Regular Cash Sale
  - a) Prompted for item name.
    - Will accept any string input.
    - '\*' input indicates end of item prompt.
      - Will reject this if this sale's vector of items has a size of 0.
      - Jumps to step (d)
  - b) Prompted for amount of item entered. Will accept any double input.
    - Any input not conforming to these constraints will initiate a re-prompt.
  - c) Item is created with info from parts (a) & (b) and placed in this sale's vector of items.
  - d) Repeat steps (a) & (b) until \* input in step (a)
  - e) Prints subtotal amount, "regular sale", tax amount, and total amount to 2 digit decimal precision.
    - All values are also stored to sale object.
  - f) Prompts for amount received
    - Accepts any double value greater than total.
    - Any input not conforming to these constraints will initiate a re-prompt.
  - g) Change is stored to Sale and outputted to user.
  - h) Sale pointer is added to system vector of sales. Menu of options is redisplayed.
2. User selects a Regular Check Sale
  - a) Prompted for item name.
    - Will accept any string input.
    - '\*' input indicates end of item prompt.
      - Will reject this if this sale's vector of items has a size of 0.
      - Jumps to step (d)
  - b) Prompted for amount of item entered. Will accept any double input.
    - Any input not conforming to these constraints will initiate a re-prompt.

- c) Item is created with info from parts (a) & (b) and placed in this sale's vector of items.
- d) Repeat steps (a) & (b) until \* input in step (a)
- e) Prints subtotal amount, "regular sale", tax amount, and total amount to 2 digit decimal precision.
  - All values are also stored to sale object.
- f) Prompts for name on check
  - Accepts any string input
  - Stores this information in the object.
- g) Prompts for Driver's License Number
  - Accepts any long long int input
  - Re-prompts if input does not conforming
- h) Sale pointer is added to system vector of sales. Menu options are redisplayed.

### 3. User selects a Regular Credit Card Sale

- a) Prompted for item name.
  - Will accept any string input.
  - '\*' input indicates end of item prompt.
    - Will reject this if this sale's vector of items has a size of 0.
    - Jumps to step (d)
- b) Prompted for amount of item entered. Will accept any double input.
  - Any input not conforming to these constraints will initiate a re-prompt.
- c) Item is created with info from parts (a) & (b) and placed in this sale's vector of items.
- d) Repeat steps (a) & (b) until \* input in step (a)
- e) Prints subtotal amount, "regular sale", tax amount, and total amount to 2 digit decimal precision.
  - All values are also stored to sale object.
- f) Prompts for credit card number.
  - Accepts any long long int input
  - Re-prompts if input doesn't conform
  - Stores this information in the object.
- g) Prompts for Driver's License Number
  - Accepts any long long int input
  - Re-prompts if input does not conforming
- h) Sale pointer is added to system vector of sales. Menu options are redisplayed.

### 4. User selects a Contract Cash Sale

- a) Prompted for item name.
  - Will accept any string input.
  - '\*' input indicates end of item prompt.
    - Will reject this if this sale's vector of items has a size of 0.

- Jumps to step (d)
- b) Prompted for amount of item entered. Will accept any double input.
- Any input not conforming to these constraints will initiate a re-prompt.
- c) Item is created with info from parts (a) & (b) and placed in this sale's vector of items.
- d) Repeat steps (a) & (b) until \* input in step (a)
- e) Prints "contractor sale" and prompts for Contractor ID
- Accepts any string and stores it in the sale as id.
- f) Prints subtotal amount, tax amount, discounted subtotal, tax, and final total amount to 2 digit decimal precision.
- New subtotal includes 15% off original subtotal. Tax calculated based on new subtotal.
  - All values are also stored to sale object.
- g) Prompts for amount received
- Accepts any double value greater than total.
  - Any input not conforming to these constraints will initiate a re-prompt.
- h) Change is stored to Sale and outputted to user.
- i) Sale pointer is added to system vector of sales. Menu of options is redisplayed.

## 5. User selects a Contractor Check Sale

- a) Prompted for item name.
- Will accept any string input.
  - '\*' input indicates end of item prompt.
    - Will reject this if this sale's vector of items has a size of 0.
    - Jumps to step (d)
- b) Prompted for amount of item entered. Will accept any double input.
- Any input not conforming to these constraints will initiate a re-prompt.
- c) Item is created with info from parts (a) & (b) and placed in this sale's vector of items.
- d) Repeat steps (a) & (b) until \* input in step (a)
- e) Prints "contract sale" and prompts for contractor ID
- Accepts any string value, stores it in "id"
- f) Prints subtotal amount, discount, new subtotal, tax amount, and final total amount to 2 digit decimal precision.
- New subtotal includes 15% off original subtotal. Tax calculated based on new subtotal.
  - All values are also stored to sale object.
- g) Prompts for name on check
- Accepts any string input
  - Stores this information in the object.
- h) Prompts for Driver's License Number

- Accepts any long long int input
  - Re-prompts if input does not conforming
- i) Sale pointer is added to system vector of sales. Menu options are redisplayed.
6. User selects a Contractor Credit Card Sale
- Prompted for item name.
    - Will accept any string input.
    - '\*' input indicates end of item prompt.
      - Will reject this if this sale's vector of items has a size of 0.
      - Jumps to step (d)
  - Prompted for amount of item entered. Will accept any double input.
    - Any input not conforming to these constraints will initiate a re-prompt.
  - Item is created with info from parts (a) & (b) and placed in this sale's vector of items.
  - Repeat steps (a) & (b) until \* input in step (a)
  - Prints "contract" sale" and prompts for contractor ID
    - Accepts any string value, stores it in "id"
  - Prints subtotal amount, "regular sale", tax amount, and total amount to 2 digit decimal precision.
    - New subtotal includes 15% off original subtotal. Tax calculated based on new subtotal.
    - All values are also stored to sale object.
  - Prompts for credit card number.
    - Accepts any long long int input
    - Re-prompts if input doesn't conform
    - Stores this information in the object.
  - Prompts for Driver's License Number
    - Accepts any long long int input
    - Re-prompts if input does not conforming
  - Sale pointer is added to system vector of sales. Menu options are redisplayed.
7. User selects to print all sales.
- Each sale is printed by iterating through system vector of Sales.
    - Starts with "Sale #" based on order entered (I.e first sale would be Sale #1)
    - Each item of sale is printed by iterating through Sale vector items.
      - Each Item is printed with index, name, and cost amount.
      - All of the information is printed associated sale (I,e subtotal, discounts, total costs, ID numbers, check numbers, credit card number, expiration date, driver license numbers as applicable)
8. User selects to quit

a) Program terminates gracefully with exit code 0