Project Vending_Machine (100 points)

Create a program that simulates a pop (soda) vending machine. This project gives insight into what it takes to design an operating system command line interpreter. **The vending machine software has two working modes: "Service mode" and "Normal mode".** The service mode is used to add inventory such as different brand of soda, cups, coins, and bills. In normal mode, the program should keep track of how much money is inserted and how much money needs to be returned to the customer. When the customer picks an option, there has to be an existing quantity for that option. As products are dispensed, the program subtracts from the machine inventory. **Initially, the program comes up in service mode**, where the service person can fill-up coins, bills, pop, cups. The machine will not dispense pop if there are no cups present.

To go to the normal mode where the customer can request a drink, the technician enters the command "Lock password" where the password is the one entered with the 'LOCK' command without the double quotes. You can define your own service password. In order for the customer to request a drink the machine should have the right inventory to be able to dispense the product that the customer chooses. To get back in the service mode to add inventory, type "Unlock password". Display "Invalid password, try again" if the wrong password is entered.

The cost of a drink is set for $0.75. If the customers insert less than this amount, display an error and prompt the user to enter more money. If the customer inserts more than this amount, dispense the change along with the chosen drink, and subtract coins, bills, cups from the inventory. The program should get the input from the command line in this form:

```
Command <Parameter>
```

For example:

```
Cola Pepsi
```

When the program starts, display the following output:

```
Please Enter a command and its parameter
(Type Help for list of commands, EXIT to quit)
[SERVICE MODE]>
```

To get to the normal mode (assume password is hard-coded as 'nancy'):

```
[SERVICE MODE]>Lock nancy
```

To get back to the service mode:

```
[NORMAL MODE]>Unlock nancy
[SERVICE MODE]>
```

Typing HELP will display the valid commands for the current mode.

```
[NORMAL MODE]>Help
Commands in Normal Mode are:
Coin <value> where value is 5 10 25 nickel dime quarter
Bill <value> where value is 1 5
Cola <value> where value is coke pepsi rc jolt faygo
Exit
Unlock [password]
```

```
[SERVICE MODE]>Help
Commands in Service Mode are:
Status
Add [COLA|CUPS] brand <quantity>
Add|Remove [Coins|Bills] <denomination> <quantity>
Exit
Lock [password]
```

The password is case sensitive. Commands and parameters are NOT case sensitive.
Type Status in Service Mode to get the status of the vending machine inventory. For example:

```
[SERVICE MODE]>Status
Amount Deposited = 0.000000
total One Dollar Bills = 0
total Five Dollar Bills = 0
total Nickels = 0
total Dimes = 0
total Quarters = 0
total Coke = 0
total Pepsi = 0
total RC = 0
total Jolt = 0
total Faygo = 0
total Cups = 0
```

Submission Requirement:
1) Whole working source code (VS C++ project solution is preferred);
2) Project report (see grading section for more detail);
3) A 5-min presentation video (MP4 format);
Put them together in project_vending_machine.zip file. This file, when unzipped should compile and run the same way on the grader's computer as the student's computer with no special instructions to the grader.

Grading

Correctness 40%: The program behaves as required per the project requirement descriptions.

Coding 20%: The program code is neat and commented; it uses object-oriented design; it contains exceptions and error handling, etc.

Report 20%: The project report is neat, readable, and well structured. The report contains (but is not limited to) sections like project introduction, design diagram, description of major data structure/algorithm/class, demonstration screenshots and description, insights, conclusions.

Presentation 20%: Present your design, explain your code, and demonstrate your program running in 5 minutes video.