Cleaning Management System

Beginner

Activity scenario summary:

In this example we will code a project which replicates the management system for a Cleaning Company.

This project will help you develop an understanding how coding is used in cleaning businesses. As part of the activity participants will design a software in which the cleaning staff will be able to;

a) View Houses to Clean
b) View Cleaned Houses
c) Book New Clients
d) Manage Houses
e) View Stocklist
f) Stock To Replenish

For the first part of this project, we will allow the user to view houses that need to be cleaned, and houses that are currently clean

coding
4 MIGRANT WOMEN RETURNERS

View Houses to Clean

```
-- Menu --
1) Houses To Clean
2) Cleaned Houses
3) Book New Client
4) Manage Houses
5) View Stocklist
6) Stock to Replenish
Enter an option:
```

View houses to clean:

- Check if there are any houses to clean
- If there are houses to clean, print out the list of houses

Find the finished code for this project here – use this if you get stuck!

coding
4 MIGRANT WOMEN RETURNERS

What will happen…

```
Enter an option: 1

--- Houses To Clean ---
1) House 1
2) House 4
3) House 5
4) House 7
5) House 9
```

Find the finished code for this project here – use this if you get stuck!

coding
4 MIGRANT WOMEN RETURNERS

Step 1

Setting up our program…

At the top of our program lets set out some variables for the data we will be storing throughout the course of the program

```
# Cleaning Management System
# --------------------------

clients = []
housesToCleanList = [1, 4, 5, 7, 9]
cleanedHousesList = [2, 3, 6]

stocklist = ["Bleach", "Detergent", "Polish", "Sanitizer", "Wipes"]
stockQuantity = [15, 75, 30, 45, 200]
stockPrice = [40, 20, 35, 65, 30]
```

Declaring variables at the top of your program is good practice, and helps keep your code organized

Step 2

Defining a subroutine and getting user input…

First we declare a subroutine - subroutines are small blocks of code that perform specific tasks (such as displaying a list of houses)

```
#View Houses To Clean
def housesToClean():
    if len(housesToCleanList) > 0:

    else:
        print("No current houses to be cleaned - try again later")
```

First, we write an IF ELSE statement that checks if the length of our housesToCleanList is greater than 0 (not empty)

If it is empty, we print out a message to the user asking them to try again later!

coding
4 MIGRANT WOMEN RETURNERS

Step 3

Displaying a list of houses…

To display the list of houses, we use a FOR loop. We go through every item in our ‘housesToCleanList’ and display its number.

```
def housesToClean():
    if len(housesToCleanList) > 0:

        print("\n--- Houses To Clean ---")
        for i in range(len(housesToCleanList)):
            print(str(i+1) + ") " + "House " + str(housesToCleanList[i]))

    else:
        print("No current houses to be cleaned - try again later")
```

Remember to label your code! It’s important to keep your user informed about what’s happening using print statements.

coding
4 MIGRANT WOMEN RETURNERS

Step 4

Setting up a menu…

For the menu, we are going to set up an infinite loop.

```
while True:
    print("\n-- Menu --")
    print("1) Houses To Clean")
    print("2) Cleaned Houses")
    print("3) Book New Client")
    print("4) Manage Houses")
    print("5) View Stocklist")
    print("6) Stock to Replenish")


    option = input("Enter an option: ")
```

Using print() statements, print out our user's options, and ask which option the user would like to perform.

coding
4 MIGRANT WOMEN RETURNERS

Step 5

Setting up a menu…

```
option = input("Enter an option: ")

if option == "1":
    housesToClean()
elif option == "2":
    pass
    #cleanedHouses()
elif option == "3":
    pass
    #bookNewClient()
elif option == "4":
    pass
    #manageHouses()
elif option == "5":
    pass
    #viewStocklist()
elif option == "6":
    pass
    #stockToReplenish()
else:
    print("Input error - try again.")
```

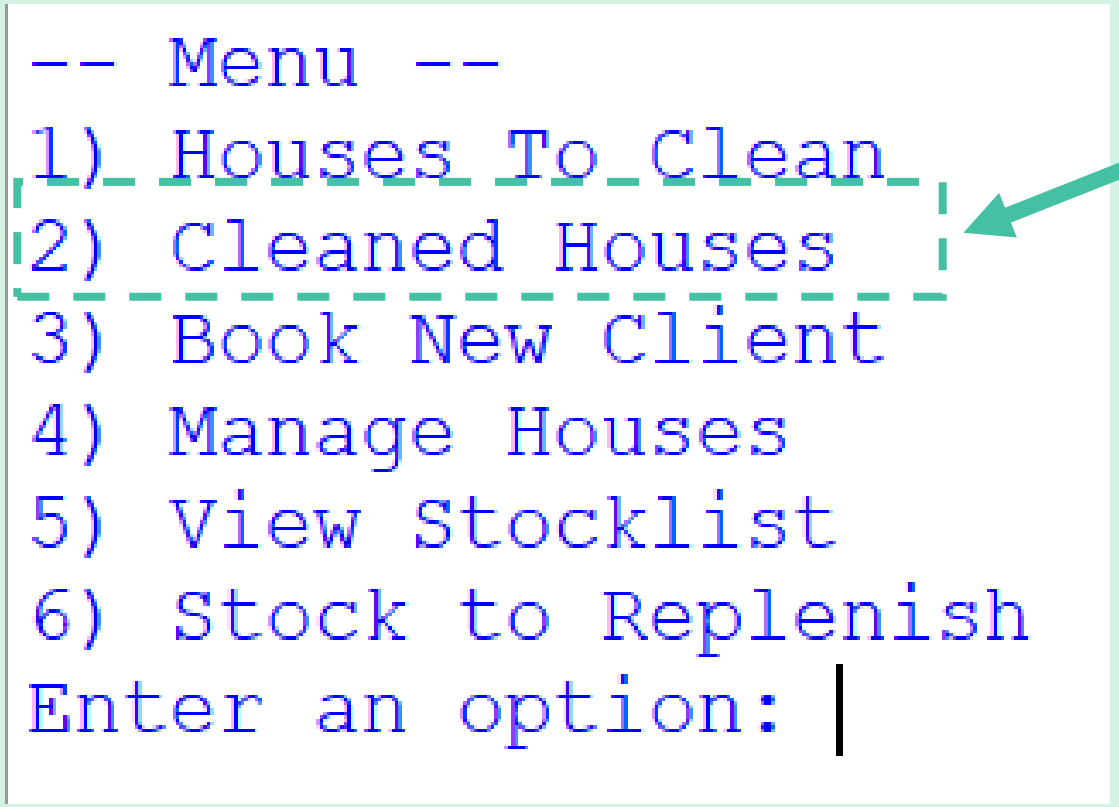
To make the menu respond to the user's input, use an IF ELIF ELSE statement. For now, we have only programmed our 'housesToClean()' subroutine, so for now, ignore the rest with pass

For each statement, check if the user has entered a number from 1-6 and call the corresponding subroutine for that option

At the bottom, have an ELSE statement that will ask them to try again if they enter something incorrectly.

coding
4 MIGRANT WOMEN RETURNERS

View Cleaned Houses



View cleaned houses:

- Check if there are any houses that have been cleaned
- If there are cleaned houses, print out the list of houses

Find the finished code for this project here – use this if you get stuck!

coding
4 MIGRANT WOMEN RETURNERS

What will happen…

```
Enter an option: 2

--- Cleaned Houses ---
1) House 2
2) House 3
3) House 6
```

Find the finished code for this project here – use this if you get stuck!

coding
4 MIGRANT WOMEN RETURNERS

Step 1

Defining a subroutine and getting user input…

Define a new subroutine and use an IF statement to check if there any items in the cleanedHousesList – if there isn't, we'll ask the user to try again later!

```
#View Cleaned Houses
def cleanedHouses():
    if len(cleanedHousesList) > 0:
        print("\n--- Cleaned Houses ---")


    else:
        print("No current clean houses - try again later")
```

It's good practice to always print out messages to the user so they know what's happening!

coding
4 MIGRANT WOMEN RETURNERS

Step 2

Display list of houses…

Use a FOR loop to go through every house in the cleanedHousesList

```
#View Cleaned Houses
def cleanedHouses():
    if len(cleanedHousesList) > 0:
        print("\n--- Cleaned Houses ---")

        for i in range(len(cleanedHousesList)):
            print(str(i+1) + ") " + "House " + str(cleanedHousesList[i]))

    else:
        print("No current clean houses - try again later")
```

For each house, print its ID and house number

```
--- Cleaned Houses ---
1) House 2
2) House 3
3) House 6
```

coding
4 MIGRANT WOMEN RETURNERS

Step 3

Update menu…

```
while True:
    print("\n-- Menu --")
    print("1) Houses To Clean")
    print("2) Cleaned Houses")
    print("3) Book New Client")
    print("4) Manage Houses")
    print("5) View Stocklist")
    print("6) Stock to Replenish")


    option = input("Enter an option: ")

    if option == "1":
        housesToClean()

    elif option == "2":
        cleanedHouses()
```

Finally, update your menu by calling the cleanedHouses() subroutine when the user enters ‘2’.

coding
4 MIGRANT WOMEN RETURNERS

What your code should look like so far…

```
# Cleaning Management System
# --------------------------

clients = []
housesToCleanList = [1, 4, 5, 7, 9]
cleanedHousesList = [2, 3, 6]

stocklist = ["Bleach", "Detergent", "Polish", "Sanitizer", "Wipes"]
stockQuantity = [15, 75, 30, 45, 200]
stockPrice = [40, 20, 35, 65, 30]

#View Houses to Clean
def housesToClean():
    if len(housesToCleanList) > 0:
        print("\n--- Houses To Clean ---")
        for i in range(len(housesToCleanList)):
            print(str(i+1) + ") " + "House " + str(housesToCleanList[i]))
    else:
        print("No current houses to be cleaned - try again later")

#View Cleaned Houses
def cleanedHouses():
    if len(cleanedHousesList) > 0:
        print("\n--- Cleaned Houses ---")

        for i in range(len(cleanedHousesList)):
            print(str(i+1) + ") " + "House " + str(cleanedHousesList[i]))

    else:
        print("No current clean houses - try again later")
```

```
while True:
    print("\n-- Menu --")
    print("1) Houses To Clean")
    print("2) Cleaned Houses")
    print("3) Book New Client")
    print("4) Manage Houses")
    print("5) View Stocklist")
    print("6) Stock to Replenish")


    option = input("Enter an option: ")

    if option == "1":
        housesToClean()

    elif option == "2":
        cleanedHouses()

    elif option == "3":
        pass
        #bookNewClient()
    elif option == "4":
        pass
        #manageHouses()
    elif option == "5":
        pass
        #viewStocklist()
    elif option == "6":
        pass
        #stockToReplenish()
    else:
        print("Input error - try again.")
```

coding
4 MIGRANT WOMEN RETURNERS

What your code should look like so far…

```
-- Menu --
1) Houses To Clean
2) Cleaned Houses
3) Book New Client
4) Manage Houses
5) View Stocklist
6) Stock to Replenish
Enter an option:
```

```
Enter an option: 1

--- Houses To Clean ---
1) House 1
2) House 4
3) House 5
4) House 7
5) House 9
```

```
Enter an option: 2

--- Cleaned Houses ---
1) House 2
2) House 3
3) House 6
```

Congratulations!

You have completed the beginner code for the cleaning software system!