Home work-2

1.Design vacuum cleaner with any 4 different shapes of you choice, Each code should at least accept “start”, “stop”, “right”, “left” and “dock”. Rest is up to your imagination. Look each shape offers unique advantage, now that’s something you need extract for your advantage and also justify why a particular shape is better

Input:

```
def smart_cleaner(shape_name, advantage, command):
    print(f"\n ~Activating {shape_name.capitalize()} shaped Vacuum Cleaner")

    # Command responses
    commands = {
        "start": f"{shape_name} shaped vacuum has begun cleaning.",
        "stop": f"The {shape_name} shaped vacuum stopped.",
        "left": f"The {shape_name} shaped vacuum is turning left.",
        "right": f"The {shape_name} shaped vacuum is turning right.",
        "dock": f"The {shape_name} shaped vacuum is returning to its docking station."
    }

    print(commands.get(command.lower(), "Command not recognized. Please try again."))
    print(f"Advantage: {advantage}")

# Main program
if __name__ == "__main__":
    print("~Welcome to Smart Vacuum Cleaner!")

    # Shape selection options with advantages
    shape_options = {
        "1": ("circle", "Smooth movement and easy turning."),
        "2": ("semi-circle", "Good for cleaning along walls."),
        "3": ("square", "Covers area efficiently."),
        "4": ("triangle", "Reaches tight corners easily."),
        "5": ("pentagon", "Stable movement in open spaces.")
    }

    print("\nSelect the vacuum shape:")
    for key, (name, _) in shape_options.items():
        print(f"{key}. {name.capitalize()}")

    shape_choice = input("Enter the number corresponding to the shape: ").strip()
    selected = shape_options.get(shape_choice)

    if not selected:
        print("Invalid shape selection. Please restart and choose a valid number.")
    else:
        command = input("Enter the command (Start, Stop, Left, Right, Dock): ").strip()
        smart_cleaner(*selected, command)
```

Output:

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python313/harsh.py =======
~Welcome to Smart Vacuum Cleaner!

Select the vacuum shape:
1. Circle
2. Semi-circle
3. Square
4. Triangle
5. Pentagon
Enter the number corresponding to the shape: 1
Enter the command (Start, Stop, Left, Right, Dock): start

 ~Activating Circle shaped Vacuum Cleaner
circle shaped vacuum has begun cleaning.
Advantage: Smooth movement and easy turning.
>>>
======== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python313/harsh.py =======
~Welcome to Smart Vacuum Cleaner!

Select the vacuum shape:
1. Circle
2. Semi-circle
3. Square
4. Triangle
5. Pentagon
Enter the number corresponding to the shape: 2
Enter the command (Start, Stop, Left, Right, Dock): left

 ~Activating Semi-circle shaped Vacuum Cleaner
The semi-circle shaped vacuum is turning left.
Advantage: Good for cleaning along walls.
...
```

```
======== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python313/harsh.py =======
~Welcome to Smart Vacuum Cleaner!

Select the vacuum shape:
1. Circle
2. Semi-circle
3. Square
4. Triangle
5. Pentagon
Enter the number corresponding to the shape: 5
Enter the command (Start, Stop, Left, Right, Dock): right

 ~Activating Pentagon shaped Vacuum Cleaner
The pentagon shaped vacuum is turning right.
Advantage: Stable movement in open spaces.
>>>
======== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python313/harsh.py =======
~Welcome to Smart Vacuum Cleaner!

Select the vacuum shape:
1. Circle
2. Semi-circle
3. Square
4. Triangle
5. Pentagon
Enter the number corresponding to the shape: 3
Enter the command (Start, Stop, Left, Right, Dock): dock

 ~Activating Square shaped Vacuum Cleaner
The square shaped vacuum is returning to its docking station.
Advantage: Covers area efficiently.
>>>
======== RESTART: C:/Users/User/AppData/Local/Programs/Python/Python313/harsh.py =======
~Welcome to Smart Vacuum Cleaner!

Select the vacuum shape:
1. Circle
2. Semi-circle
3. Square
4. Triangle
5. Pentagon
Enter the number corresponding to the shape: 4
Enter the command (Start, Stop, Left, Right, Dock): stop

 ~Activating Triangle shaped Vacuum Cleaner
The triangle shaped vacuum stopped.
Advantage: Reaches tight corners easily.
```

Code:

```
def smart_cleaner(shape_name, advantage, command):
    print(f"\n ~Activating {shape_name.capitalize()} shaped Vacuum Cleaner")

    # Command responses
    commands = {
        "start": f"{shape_name} shaped vacuum has begun cleaning.",
        "stop": f"The {shape_name} shaped vacuum stopped.",
        "left": f"The {shape_name} shaped vacuum is turning left.",
        "right": f"The {shape_name} shaped vacuum is turning right.",
        "dock": f"The {shape_name} shaped vacuum is returning to its docking station."
    }

    print(commands.get(command.lower(), "Command not recognized. Please try again."))
    print(f'Advantage: {advantage}")

# Main program
if _name_ == "_main_":
    print("~Welcome to Smart Vacuum Cleaner!")

    # Shape selection options with advantages
    shape_options = {
        "1": ("circle", "Smooth movement and easy turning."),
        "2": ("semi-circle", "Good for cleaning along walls."),
        "3": ("square", "Covers area efficiently."),
        "4": ("triangle", "Reaches tight corners easily."),
        "5": ("pentagon", "Stable movement in open spaces.")
    }
```

```
print("\nSelect the vacuum shape:")
for key, (name, _) in shape_options.items():
    print(f"{key}. {name.capitalize()}")

shape_choice = input("Enter the number corresponding to the shape: ").strip()
selected = shape_options.get(shape_choice)

if not selected:
    print("Invalid shape selection. Please restart and choose a valid number.")
else:
    command = input("Enter the command (Start, Stop, Left, Right, Dock): ").strip()
    smart_cleaner(*selected, command)
```