EXPERIMENT NO. 1

Student Name and Roll Number: Piyush Gambhir – 21CSU349
Semester /Section: Semester-V – AIML-V-B (AL-3)
Link to Code: NCU-Lab-Manual-And-End-Semester-Projects/NCU-CSL347-AAIES-Lab_Manual at main · Piyush-Gambhir/NCU-Lab-Manual-And-End-Semester-Projects (github.com)
Date: 05.08.2023
Faculty Signature:
Grade:

Objective(s):

- Understand what a rule-based system is.
- Study about how rule-based systems work.
- Design a rule-based system for solving a real-world problem.

Outcome:

Students would be able to design Rule based systems for solving problems.

Problem Statement:

Develop a rule-based pet care assistant in Python with the purpose of providing users with helpful reminders and suggestions for pet care.
The program will offer personalized care tips based on the type of pet, including dogs, cats, and birds, using predefined rules.

Background Study: Rule-based decision-making is a fundamental concept in computer science and artificial intelligence. It involves defining a set of rules or conditions to guide the decision-making process. In the context of the pet care assistant, these rules will be used to determine the type of pet and provide relevant care tips based on the user's selection.

Question Bank:

1. What are some challenges that you faced when creating this rule-based system?
   - Knowledge Acquisition: Gathering accurate and comprehensive rules from experts can be time-consuming and challenging.
   - Rule Complexity: Dealing with a large number of rules and managing their interactions can be complex.
   - Maintenance: Regular updates and adjustments to rules are necessary as knowledge evolves.
   - Knowledge Elicitation: Expressing expert knowledge in a machine-readable format can be prone to misinterpretation.
   - Handling Exceptions: Dealing with exceptions and handling edge cases within rules can be difficult.

2. What could be some advantages and disadvantages of such Rule based systems?
   Advantages:
   - **Transparency**: The reasoning behind decisions is explicit and interpretable.

- **Structured Knowledge**: Rules organize domain knowledge in a structured manner.
- **Control**: Experts can directly define and modify rules, providing control over decision-making.
- **Scalability**: Rule-based systems can handle a wide range of cases by adding or modifying rules.

Disadvantages:

- **Limited Context**: May struggle with nuanced decision-making that requires broader context or common sense.
- **Brittleness**: Highly dependent on accurate rules and can fail when rules conflict or are incomplete.
- **Maintenance Overhead**: Regular updates and adjustments to rules can be resource-intensive.
- **Lack of Learning**: Rule-based systems often lack the ability to adapt and learn from new data.

3. How can rule-based systems be made more efficient?
   - Rule Pruning: Eliminate redundant or less impactful rules to streamline decision-making.
   - Rule Ordering: Arrange rules strategically to prioritize more important conditions.
   - Caching and Memoization: Store intermediate results to avoid redundant calculations.
   - Fuzzy Logic: Incorporate fuzzy logic to handle imprecise or uncertain information.
   - Inference Optimization: Use efficient algorithms for rule matching and inference, like Rete algorithm.
   - Machine Learning Integration: Combine rule-based systems with machine learning to learn patterns from data.
   - Hybrid Approaches: Integrate rule-based systems with probabilistic or neural network models for enhanced accuracy and robustness.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Approach

1. **get_pet_choice()**: This function displays pet options to the user (Dog, Cat, Parrot) and prompts the user to select a pet by entering a corresponding number. It ensures the user's input is valid and then returns the choice.

2. **get_pet_age()**: This function prompts the user to enter their pet's age in years. It ensures the age is a valid non-negative number and then returns it.

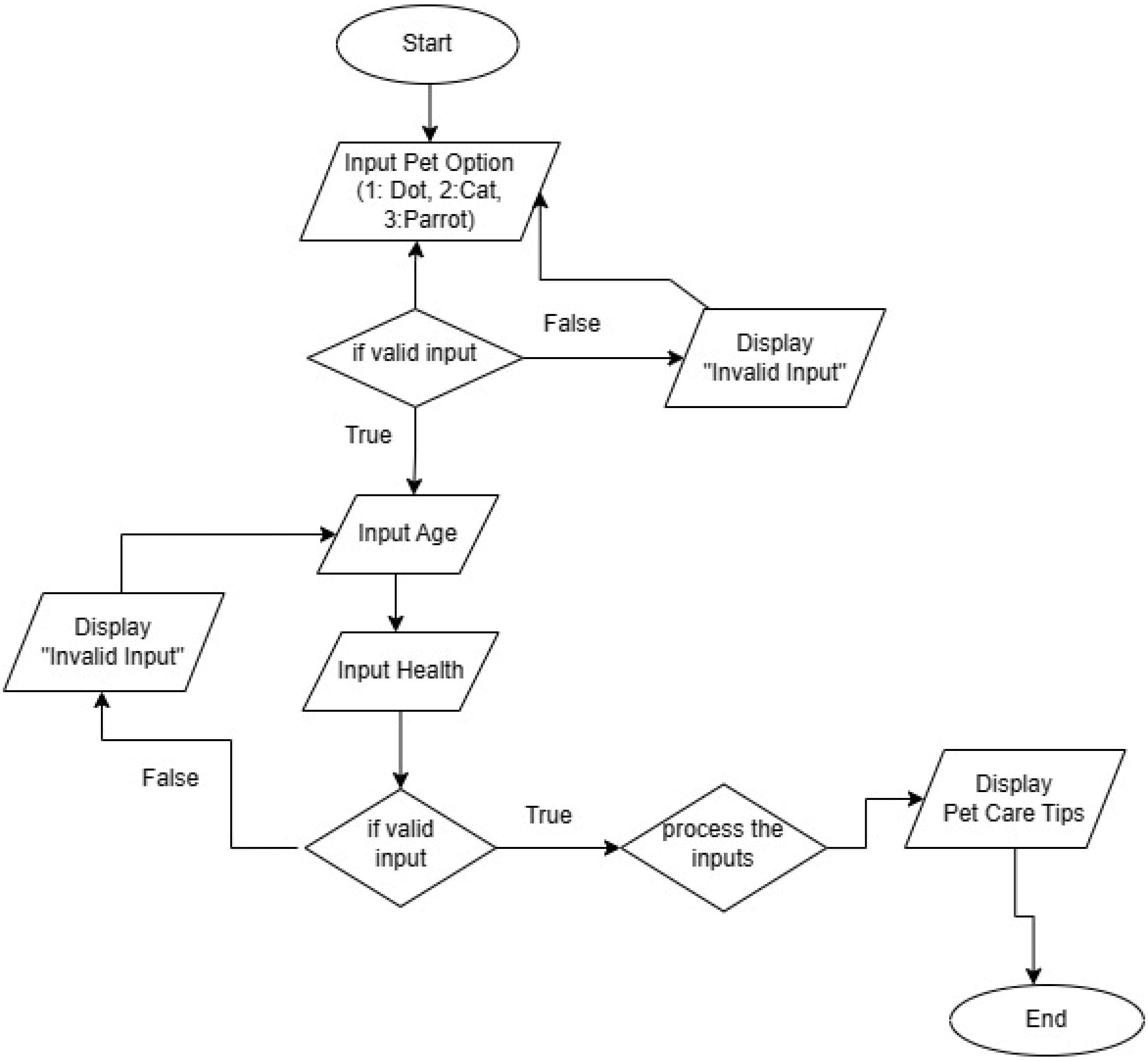
3. **get_pet_health()**: This function prompts the user to select their pet's health status (Good, Fair, Poor) by entering a corresponding number. It ensures the user's input is valid and then returns the health status as a string.

4. **provide_pet_care_tips()**: This function provides pet care tips based on the pet's type (Dog, Cat, Parrot), age, and health status.

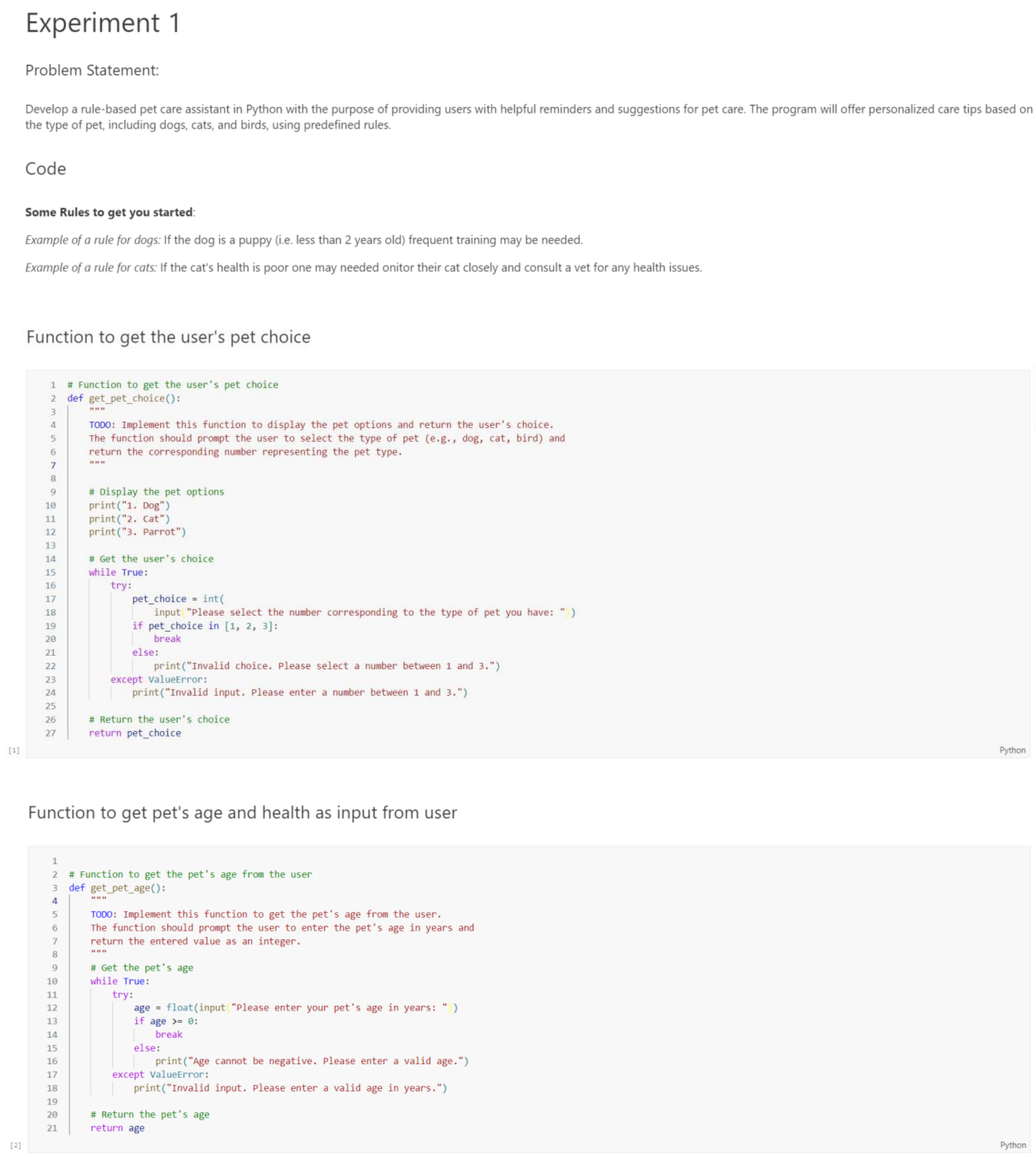
5. **pet_care_assistant()**: This is the main function that serves as the entry point for the pet care assistant program. It calls the previous three functions (**get_pet_choice()**, **get_pet_age()**, and **get_pet_health()**) to collect necessary information from the user. Then, it calls the **provide_pet_care_tips()** function to provide personalized care tips based on the user's inputs.

6. **__main__ block**: The program starts by calling the **pet_care_assistant()** function.

Flowchart


Start
Input Pet Option (1: Dot, 2:Cat, 3:Parrot)
if valid input
False
Display "Invalid Input"
True
Input Age
Display "Invalid Input"
Input Health
False
if valid input
True
process the inputs
Display Pet Care Tips
End

Code



Experiment 1

Problem Statement:

Develop a rule-based pet care assistant in Python with the purpose of providing users with helpful reminders and suggestions for pet care. The program will offer personalized care tips based on the type of pet, including dogs, cats, and birds, using predefined rules.

Code

Some Rules to get you started:

Example of a rule for dogs: If the dog is a puppy (i.e. less than 2 years old) frequent training may be needed.

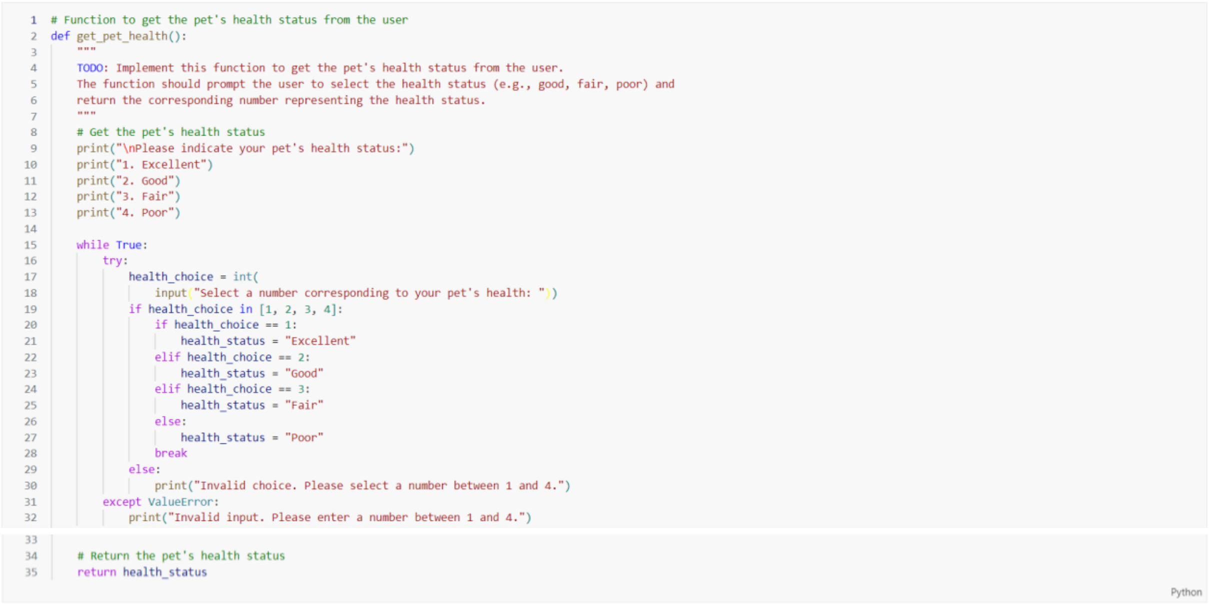
Example of a rule for cats: If the cat's health is poor one may needed onitor their cat closely and consult a vet for any health issues.

Function to get the user's pet choice

```python
# Function to get the user's pet choice
def get_pet_choice():
    """
    TODO: Implement this function to display the pet options and return the user's choice.
    The function should prompt the user to select the type of pet (e.g., dog, cat, bird) and
    return the corresponding number representing the pet type.
    """

    # Display the pet options
    print("1. Dog")
    print("2. Cat")
    print("3. Parrot")

    # Get the user's choice
    while True:
        try:
            pet_choice = int(
                input("Please select the number corresponding to the type of pet you have: "))
            if pet_choice in [1, 2, 3]:
                break
            else:
                print("Invalid choice. Please select a number between 1 and 3.")
        except ValueError:
            print("Invalid input. Please enter a number between 1 and 3.")

    # Return the user's choice
    return pet_choice
```

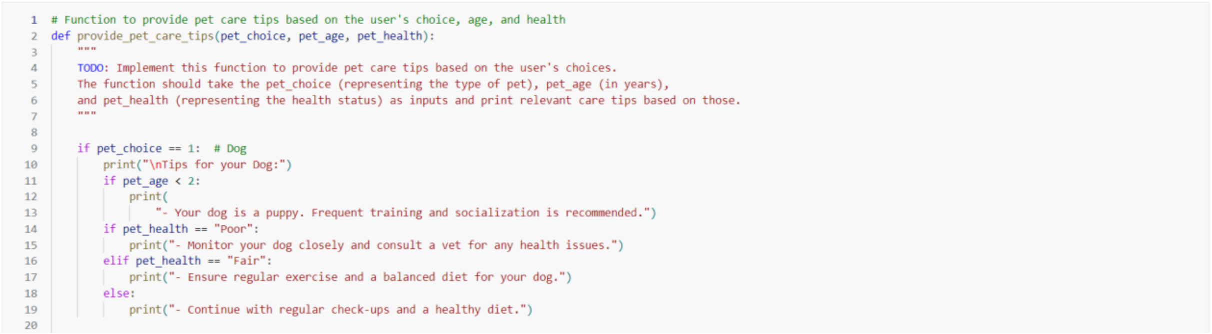
Function to get pet's age and health as input from user

```python

# Function to get the pet's age from the user
def get_pet_age():
    """
    TODO: Implement this function to get the pet's age from the user.
    The function should prompt the user to enter the pet's age in years and
    return the entered value as an integer.
    """
    # Get the pet's age
    while True:
        try:
            age = float(input("Please enter your pet's age in years: "))
            if age >= 0:
                break
            else:
                print("Age cannot be negative. Please enter a valid age.")
        except ValueError:
            print("Invalid input. Please enter a valid age in years.")

    # Return the pet's age
    return age
```

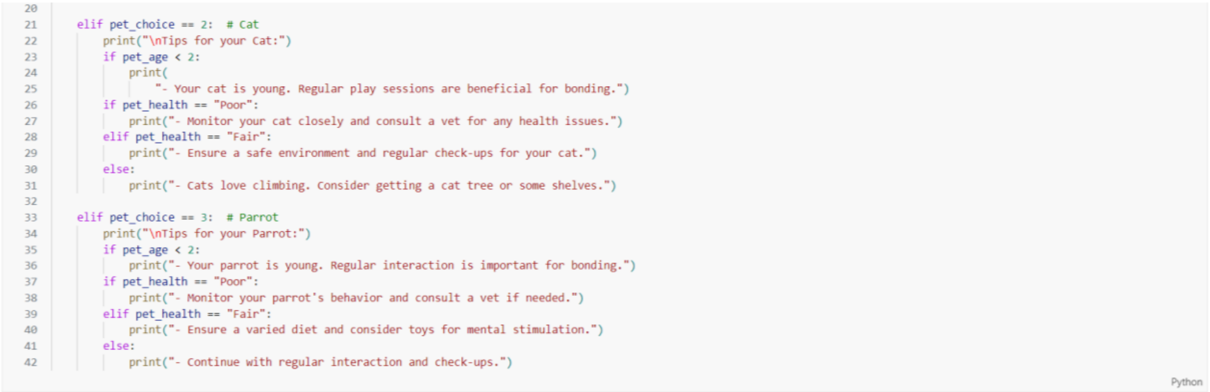


```python
# Function to get the pet's health status from the user
def get_pet_health():
    """
    TODO: Implement this function to get the pet's health status from the user.
    The function should prompt the user to select the health status (e.g., good, fair, poor) and
    return the corresponding number representing the health status.
    """
    # Get the pet's health status
    print("\nPlease indicate your pet's health status:")
    print("1. Excellent")
    print("2. Good")
    print("3. Fair")
    print("4. Poor")

    while True:
        try:
            health_choice = int(
                input("Select a number corresponding to your pet's health: "))
            if health_choice in [1, 2, 3, 4]:
                if health_choice == 1:
                    health_status = "Excellent"
                elif health_choice == 2:
                    health_status = "Good"
                elif health_choice == 3:
                    health_status = "Fair"
                else:
                    health_status = "Poor"
                break
            else:
                print("Invalid choice. Please select a number between 1 and 4.")
        except ValueError:
            print("Invalid input. Please enter a number between 1 and 4.")

    # Return the pet's health status
    return health_status
```

Function for rules for providing pet care tips



```python
# Function to provide pet care tips based on the user's choice, age, and health
def provide_pet_care_tips(pet_choice, pet_age, pet_health):
    """
    TODO: Implement this function to provide pet care tips based on the user's choices.
    The function should take the pet_choice (representing the type of pet), pet_age (in years),
    and pet_health (representing the health status) as inputs and print relevant care tips based on those.
    """

    if pet_choice == 1:  # Dog
        print("\nTips for your Dog:")
        if pet_age < 2:
            print(
                "- Your dog is a puppy. Frequent training and socialization is recommended.")
        if pet_health == "Poor":
            print("- Monitor your dog closely and consult a vet for any health issues.")
        elif pet_health == "Fair":
            print("- Ensure regular exercise and a balanced diet for your dog.")
        else:
            print("- Continue with regular check-ups and a healthy diet.")

```



```python

    elif pet_choice == 2:  # Cat
        print("\nTips for your Cat:")
        if pet_age < 2:
            print(
                "- Your cat is young. Regular play sessions are beneficial for bonding.")
        if pet_health == "Poor":
            print("- Monitor your cat closely and consult a vet for any health issues.")
        elif pet_health == "Fair":
            print("- Ensure a safe environment and regular check-ups for your cat.")
        else:
            print("- Cats love climbing. Consider getting a cat tree or some shelves.")

    elif pet_choice == 3:  # Parrot
        print("\nTips for your Parrot:")
        if pet_age < 2:
            print("- Your parrot is young. Regular interaction is important for bonding.")
        if pet_health == "Poor":
            print("- Monitor your parrot's behavior and consult a vet if needed.")
        elif pet_health == "Fair":
            print("- Ensure a varied diet and consider toys for mental stimulation.")
        else:
            print("- Continue with regular interaction and check-ups.")
```

Main function to run the pet care assistant

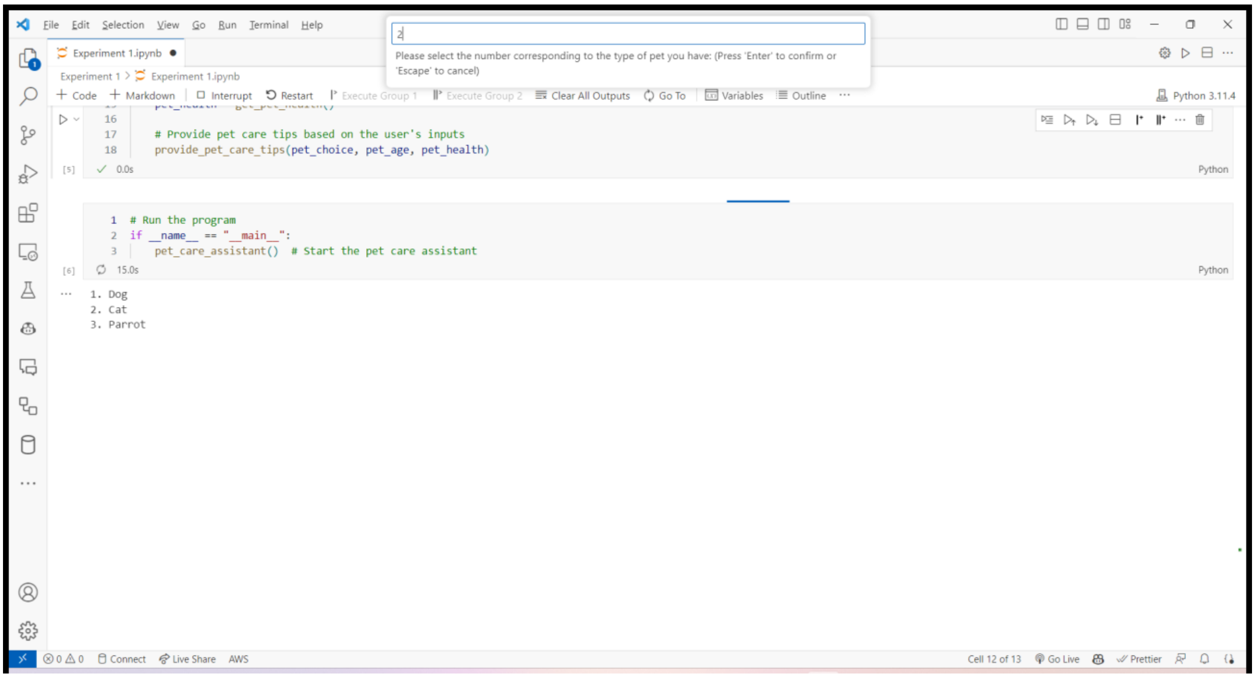
```python
# Main function to run the pet care assistant
def pet_care_assistant():
    """
    TODO: Implement this function as the entry point to the pet care assistant program.
    The function should call the get_pet_choice(), get_pet_age(), and get_pet_health() functions
    to collect the necessary information from the user. Then, it should call the provide_pet_care_tips()
    function to provide personalized care tips based on the user's inputs.
    """

    # Get pet choice from the user
    pet_choice = get_pet_choice()

    # Get pet details (age and health status) from the user
    pet_age = get_pet_age()
    pet_health = get_pet_health()

    # Provide pet care tips based on the user's inputs
    provide_pet_care_tips(pet_choice, pet_age, pet_health)
```

[5] Python

```python
# Run the program
if __name__ == "__main__":
    pet_care_assistant()  # Start the pet care assistant
```

[6] Python

Sample Output



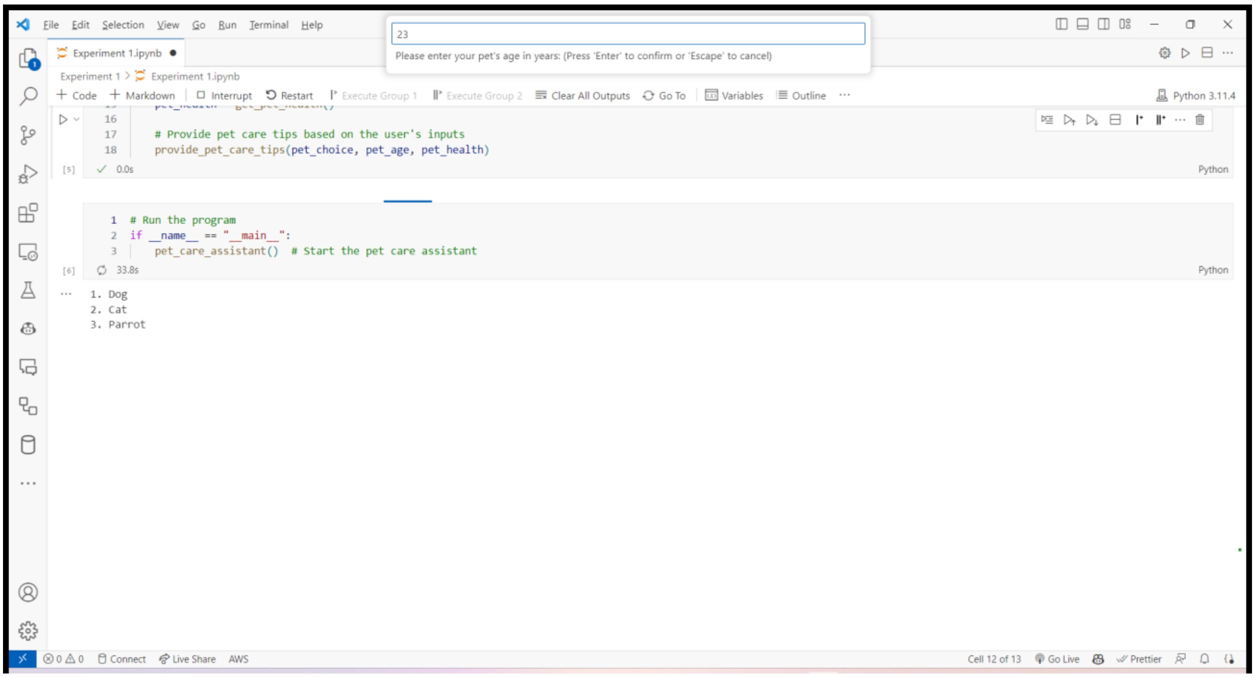
File Edit Selection View Go Run Terminal Help
Experiment 1.ipynb
Experiment 1 > Experiment 1.ipynb
+ Code + Markdown Interrupt Restart Execute Group 1 Execute Group 2 Clear All Outputs Go To Variables Outline
Python 3.11.4
Python
Cell 12 of 13 Go Live Prettier
Connect Live Share AWS

```
2
Please select the number corresponding to the type of pet you have: (Press 'Enter' to confirm or
'Escape' to cancel)
```

```

    # Provide pet care tips based on the user's inputs
    provide_pet_care_tips(pet_choice, pet_age, pet_health)
[5]  ✓ 0.0s
```

```
# Run the program
if __name__ == "__main__":
    pet_care_assistant()  # Start the pet care assistant
[6]  15.0s
```

```
1. Dog
2. Cat
3. Parrot
```



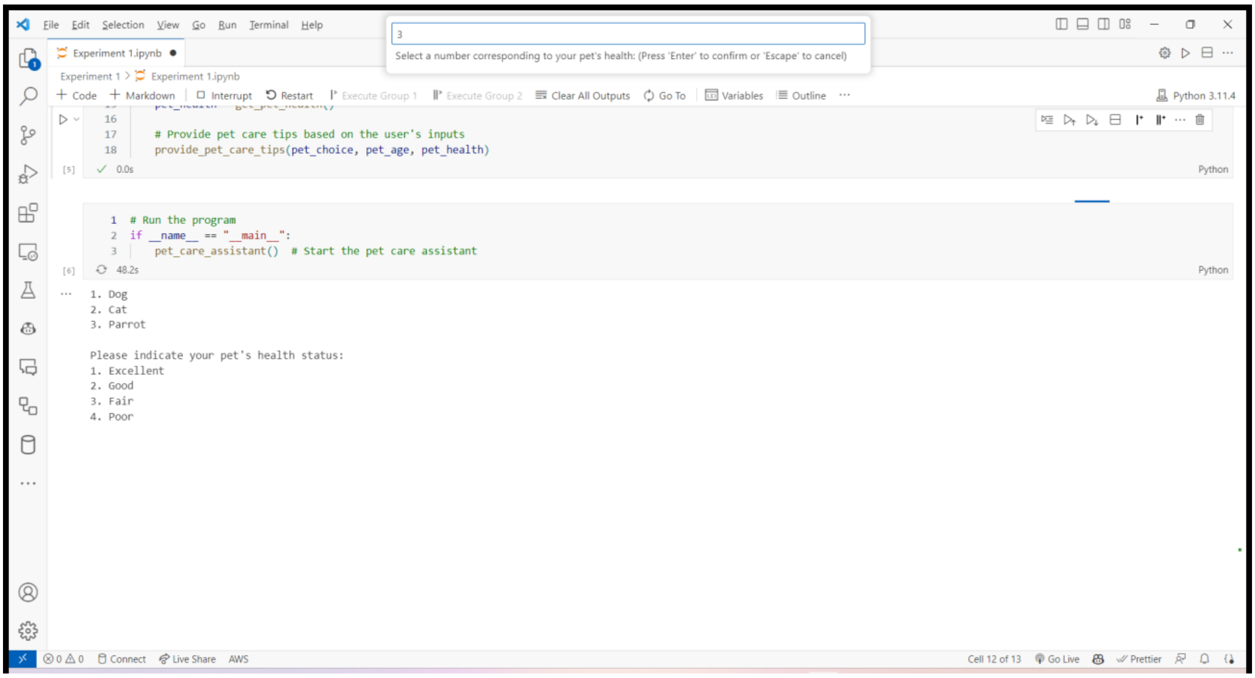
File Edit Selection View Go Run Terminal Help
Experiment 1.ipynb
Experiment 1 > Experiment 1.ipynb
+ Code + Markdown Interrupt Restart Execute Group 1 Execute Group 2 Clear All Outputs Go To Variables Outline
Python 3.11.4
Python
Cell 12 of 13 Go Live Prettier
Connect Live Share AWS

```
23
Please enter your pet's age in years: (Press 'Enter' to confirm or 'Escape' to cancel)
```

```

    # Provide pet care tips based on the user's inputs
    provide_pet_care_tips(pet_choice, pet_age, pet_health)
[5]  ✓ 0.0s
```

```
# Run the program
if __name__ == "__main__":
    pet_care_assistant()  # Start the pet care assistant
[6]  33.8s
```

```
1. Dog
2. Cat
3. Parrot
```


File Edit Selection View Go Run Terminal Help
Experiment 1.ipynb
Experiment 1 > Experiment 1.ipynb
+ Code + Markdown Interrupt Restart Execute Group 1 Execute Group 2 Clear All Outputs Go To Variables Outline
Python 3.11.4
Cell 12 of 13 Go Live Prettier
Connect Live Share AWS

```
3
```

Select a number corresponding to your pet's health: (Press 'Enter' to confirm or 'Escape' to cancel)

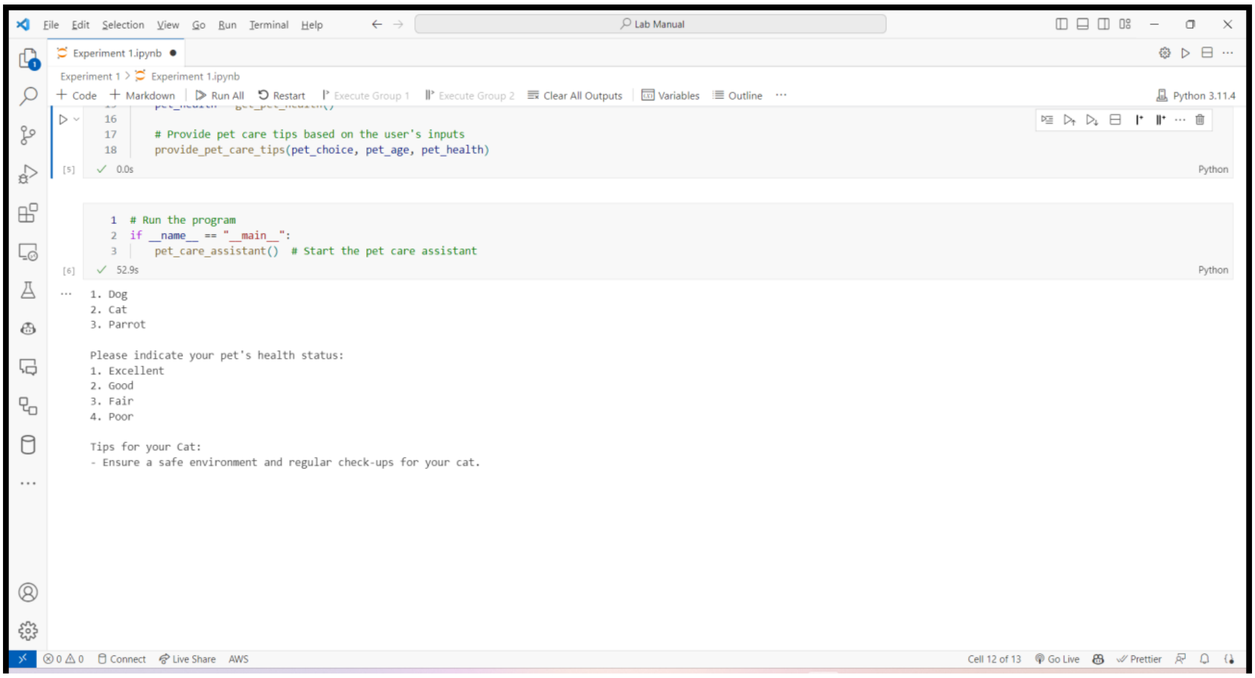
```python
    pet_health = get_pet_health()

    # Provide pet care tips based on the user's inputs
    provide_pet_care_tips(pet_choice, pet_age, pet_health)
```

[5] ✓ 0.0s

```python
# Run the program
if __name__ == "__main__":
    pet_care_assistant()  # Start the pet care assistant
```

[6] 48.2s

```
1. Dog
2. Cat
3. Parrot

Please indicate your pet's health status:
1. Excellent
2. Good
3. Fair
4. Poor
```


File Edit Selection View Go Run Terminal Help
Lab Manual
Experiment 1.ipynb
Experiment 1 > Experiment 1.ipynb
+ Code + Markdown Run All Restart Execute Group 1 Execute Group 2 Clear All Outputs Variables Outline
Python 3.11.4
Cell 12 of 13 Go Live Prettier
Connect Live Share AWS

```python
    pet_health = get_pet_health()

    # Provide pet care tips based on the user's inputs
    provide_pet_care_tips(pet_choice, pet_age, pet_health)
```

[5] ✓ 0.0s

```python
# Run the program
if __name__ == "__main__":
    pet_care_assistant()  # Start the pet care assistant
```

[6] ✓ 52.9s

```
1. Dog
2. Cat
3. Parrot

Please indicate your pet's health status:
1. Excellent
2. Good
3. Fair
4. Poor

Tips for your Cat:
- Ensure a safe environment and regular check-ups for your cat.
```