

# Pseudocode for driver.py (ENHANCED)

```
FROM typing IMPORT Dict
FROM dog.py IMPORT Dog
FROM monkey.py IMPORT Monkey
FROM reserve_animal.py IMPORT ReserveAnimal
IMPORT intake.py
IMPORT initialize.py
IMPORT print_animals.py
IMPORT search.py

CLASS Driver:
    INIT dog_dict DICTIONARY
    INIT monkey_dict DICTIONARY

    METHOD main()
        INIT scnr AS input
        FROM initialize INIT initialize_dog_dict
        FROM initialize INIT initialize_monkey_dict

        SET usr_input TO NOTHING

        WHILE usr_input IS NOT q
            DISPLAY menu
            SET usr_input TO scnr

            IF usr_input EQUALS 1
                DISPLAY "Beginning Dog Intake"
                FROM intake CALL intake_new_dog

            ELSE IF usr_input EQUALS 2
                DISPLAY "Beginning Monkey Intake"
                FROM intake CALL intake_new_monkey

            ELSE IF usr_input EQUALS 3
                DISPLAY "Beginning Animal Reservation"
                FROM ReserveAnimal CALL reserve_animal

            ELSE IF usr_input EQUALS 4
                DISPLAY "Displaying List of dogs"
                FROM print_animals CALL print_animals(1)

            ELSE IF usr_input EQUALS 5
                DISPLAY "Displaying List of monkeys"
                FROM print_animals CALL print_animals(2)

            ELSE IF usr_input EQUALS 6
                DISPLAY "Displaying List of unreserved animals"
                FROM print_animals CALL print_animals(3)

            ELSE IF usr_input EQUALS 7
                DISPLAY "Searching Database"
                FROM search CALL search_animal()

            ELSE IF usr_input EQUALS q
```

```

        DISPLAY "Quitting."
        END

    ELSE
        DISPLAY "Invalid command."

    END IF
END WHILE

METHOD display_menu()
    DISPLAY 2 new line
    DISPLAY "Rescue Animal System Menu"
    DISPLAY "[1] Intake a new dog"
    DISPLAY "[2] Intake a new monkey"
    DISPLAY "[3] Reserve an animal"
    DISPLAY "[4] Print a list of all dogs"
    DISPLAY "[5] Print a list of all monkeys"
    DISPLAY "[6] Print a list of all animals that are not reserved"
    DISPLAY "[7] Search for animal, and update status"
    DISPLAY "[q] Quit application"
    DISPLAY "Enter a menu selection"

```

## Pseudocode for dog.py

```
FROM rescueanimal.py IMPORT RescueAnimal
```

```
CLASS Dog
```

```
    Constructor METHOD (self, name, breed, gender, age, weight, acquisition_date,
acquisition_country, training_status, reserved, in_service_country)
```

```
        CALL super() METHOD(name, gender, age, weight, acquisition_date,
acquisition_country, training_status, reserved, in_service_country)
```

```
        SET self.name TO name
```

```
        SET self.breed TO breed
```

```
        SET self.gender TO gender
```

```
        SET self.age TO age
```

```
        SET self.weight TO weight
```

```
        SET self.acquisition_date TO acquisition_date
```

```
        SET self.acquisition_country TO acquisition_country
```

```
        SET self.training_status TO training_status
```

```
        SET self.reserved TO reserved
```

```
        SET self.in_service_country TO in_service_country
```

```
    METHOD get_breed(self)
```

```
        RETURN self.breed
```

```
    METHOD set_breed(self, breed)
```

```
        SET self.breed TO breed
```

## Pseudocode for monkey.py

```
FROM rescueanimal.py IMPORT RescueAnimal
```

```
CLASS Monkey(RescueAnimal)
```

```
    Constructor METHOD (self, name, species, gender, age, weight, tail_length, height, body_length,  
torso_length, skull_length, neck_length, acquisition_date, acquisition_country, training_status, reserved,  
in_service_country)
```

```
        CALL super() METHOD(name, gender, age, weight, acquisition_date,  
acquisition_country, training_status, reserved, in_service_country)
```

```
        SET self.name TO name
```

```
        SET self.species TO species
```

```
        SET self.gender TO gender
```

```
        SET self.age TO age
```

```
        SET self.weight TO weight
```

```
        SET self.tail_length TO tail_length
```

```
        SET self.height TO height
```

```
        SET self.body_length TO body_length
```

```
        SET self.torso_length TO torso_length
```

```
        SET self.skull_length TO skull_length
```

```
        SET self.neck_length TO neck_length
```

```
        SET self.acquisition_date TO acquisition_date
```

```
        SET self.acquisition_country TO acquisition_country
```

```
        SET self.training_status TO training_status
```

```
        SET self.reserved TO reserved
```

```
        SET self.in_service_country TO in_service_country
```

```
METHOD get_species(self)
```

```
    RETURN self.species
```

```
METHOD set_species(self, species)
```

```
    SET self.species TO species
```

```
METHOD get_tail_length(self)
```

```
    RETURN self.tail_length
```

```
METHOD set_tail_length(self, tail_length)
```

```
    SET self.tail_length TO tail_length
```

```
METHOD get_height(self)
```

```
    RETURN self.height
```

```
METHOD set_height(self, height)
```

```
    SET self.height TO height
```

```
METHOD get_body_length(self)
```

```
    RETURN self.body_length
```

```
METHOD set_body_length(self, body_length)
```

```
    SET self.body_length TO body_length
```

```
METHOD get_torso_length(self)
```

```
    RETURN self.torso_length
```

```
METHOD set_torso_length(self, torso_length)
```

```
    SET self.torso_length TO torso_length
```

```
METHOD get_skull_length(self)
```

```
    RETURN self.skull_length
```

```
METHOD set_skull_length(self, skull_length)
    SET self.skull_length TO skull_length
```

```
METHOD get_neck_length(self)
    RETURN self.neck_length
```

```
METHOD set_neck_length(self, neck_length)
    SET self.neck_length TO neck_length
```

## Pseudocode for rescue\_animal.py

```
CLASS RescueAnimal
```

```
    Constructor METHOD (self, name, gender, age, weight, acquisition_date, acquisition_country,
training_status, reserved, in_service_country)
```

```
        SET self.name TO name
        SET self.gender TO gender
        SET self.age TO age
        SET self.weight TO weight
        SET self.acquisition_date TO acquisition_date
        SET self.acquisition_country TO acquisition_country
        SET self.training_status TO training_status
        SET self.reserved TO reserved
        SET self.in_service_country TO in_service_country
```

```
METHOD get_name(self)
    RETURN self.name
```

```
METHOD set_name(self, name)
    SET self.name TO name
```

```
METHOD get_gender(self)
    RETURN self.gender
```

```
METHOD set_gender(self, gender)
    SET self.gender TO gender
```

```
METHOD get_age(self)
    RETURN self.age
```

```
METHOD set_age(self, age)
    SET self.age TO age
```

```
METHOD get_weight(self)
    RETURN self.weight
```

```
METHOD set_weight(self, weight)
    SET self.weight TO weight
```

```
METHOD get_acquisition_date(self)
    RETURN self.acquisition_date
```

```
METHOD set_acquisition_date(self, acquisition_date)
    SET self.acquisition_date TO acquisition_date
```

```

METHOD get_acquisition_location(self)
    RETURN self.acquisition_country

METHOD set_acquisition_location(self, acquisition_country)
    SET self.acquisition_country TO acquisition_country

METHOD get_reserved(self)
    RETURN self.reserved

METHOD set_reserved(self, reserved)
    SET self.reserved TO reserved

METHOD get_in_service_location(self)
    RETURN self.in_service_country

METHOD set_in_service_country(self, in_service_country)
    SET self.in_service_country TO in_service_country

METHOD get_training_status(self)
    RETURN self.training_status

METHOD set_training_status(self, training_status)
    SET self.training_status TO training_status

```

## Pseudocode for intake.py (ENHANCED)

```

FROM dog.py IMPORT Dog
FROM monkey.py IMPORT Monkey
FROM validation IMPORT Validation

FUNCTION intake_new_dog(dog_dict)
    DISPLAY "What is the dog's name?"
    READ input ASSIGN TO name
    FROM Validation INIT null_validation ASSIGN TO name

    FOR EACH dog in dog_dict
        IF dog.name EQUALS TO name
            DISPLAY "This dog is already in our system"
            RETURN
        END IF
    END FOR
    DISPLAY Adding "Dog:" ADD name

    DISPLAY "What is the dog's breed?"
    READ input ASSIGN breed")
    FROM Validation INIT null_validation ASSIGN TO breed
    DISPLAY "Breed:" ADD breed

    DISPLAY "What is the dog's gender?"
    READ input ASSIGN gender
    FROM Validation INIT gender_validation ASSIGN TO gender
    DISPLAY "Gender:" ADD gender

```

```
DISPLAY "What is the dog's age?"
    READ input ASSIGN age
FROM Validation INIT positive_digit_validation ASSIGN TO age
DISPLAY "Age:" ADD age
```

```
DISPLAY "What is the dog's weight?"
    READ input ASSIGN weight
FROM Validation INIT float_validation ASSIGN TO weight
DISPLAY "Weight:" ADD weight
```

```
DISPLAY "When was the dog acquired? Format: mm-dd-yyyy"
    READ input ASSIGN acquisition_date
FROM Validation INIT date_validation ASSIGN TO acquisition_date
DISPLAY "Dog Acquired:" ADD acquisition_date
```

```
DISPLAY "What Country did the dog come from?"
    READ input ASSIGN acquisition_country
FROM Validation INIT null_validation ASSIGN TO acquisition_country
DISPLAY "Country of Origin:" ADD acquisition_country
```

```
DISPLAY "What is the dog's training status?"
    READ input ASSIGN training_status
FROM Validation INIT status_validation ASSIGN TO training_status
DISPLAY "Training Status:" ADD training_status
```

```
DISPLAY "Has this dog been reserved? (True or False)"
    READ input ASSIGN reserved
FROM Validation INIT boolean_validation ASSIGN TO reserved
DISPLAY "Reservation Status:" ADD reserved
```

```
DISPLAY "What country is the dog in service?"
    READ input ASSIGN in_service_country
FROM Validation INIT null_validation ASSIGN TO in_service_country
DISPLAY "Country of Service:" ADD in_service_country
```

```
CREATE dog AS Dog ASSIGN name, breed, gender, age, weight, acquisition_date,
acquisition_country, training_status, reserved, in_service_country
```

```
APPEND dog TO dog_dict
DISPLAY "New Dog" ADD name ADD "added!"
```

```
FUNCTION intake_new_monkey(monkey_dict)
    DISPLAY "What is the Monkey's name?"
    READ input ASSIGN name
FROM Validation INIT null_validation ASSIGN TO name

    FOR EACH monkey in monkey_dict
        IF monkey.name EQUALS TO name
            DISPLAY "This Monkey is already in our system"
        END IF
    RETURN
END FOR
DISPLAY "Adding Monkey:" ADD name
```

```
DISPLAY "What is the Monkey's species?"  
    READ input ASSIGN species  
FROM Validation INIT species_validation ASSIGN TO species  
DISPLAY "Species:" ADD species
```

```
DISPLAY "What is the Monkey's gender?"  
    READ input ASSIGN gender  
FROM Validation INIT gender_validation ASSIGN TO gender  
DISPLAY "Gender:" ADD gender
```

```
DISPLAY "What is the Monkey's age?"  
    READ input ASSIGN age  
FROM Validation INIT positive_digit_validation ASSIGN TO age  
DISPLAY "Age:" ADD age
```

```
DISPLAY "What is the Monkey's weight?"  
    READ input ADD weight  
FROM Validation INIT float_validation ASSIGN TO weight  
DISPLAY "Weight:" ADD weight
```

```
DISPLAY "What is the Monkey's tail length?"  
    READ input ADD tail_length  
FROM Validation INIT positive_digit_validation ASSIGN TO tail_length  
DISPLAY "Tail Length:" ADD tail_length
```

```
DISPLAY "What is the Monkey's height?"  
    READ input ASSIGN height  
FROM Validation INIT positive_digit_validation ASSIGN TO height  
DISPLAY "Height:" ADD height
```

```
DISPLAY "What is the Monkey's body length?"  
    READ input ASSIGN body_length  
FROM Validation INIT positive_digit_validation ASSIGN TO body_length  
DISPLAY "Body Length:" ADD body_length
```

```
DISPLAY "What is the Monkey's torso length?"  
    READ input ASSIGN torso_length  
FROM Validation INIT positive_digit_validation ASSIGN TO torso_length  
DISPLAY "Torso Length:" ADD torso_length
```

```
DISPLAY "What is the Monkey's skull length?"  
    READ input ASSIGN skull_length  
FROM Validation INIT positive_digit_validation ASSIGN TO skull_length  
DISPLAY "Skull Length:" ADD skull_length
```

```
DISPLAY "What is the Monkey's neck length?"  
    READ input ASSIGN neck_length  
FROM Validation INIT positive_digit_validation ASSIGN TO neck_length  
DISPLAY "Neck Length:" ADD neck_length
```

```
DISPLAY "When was the Monkey acquired? Format: mm-dd-yyyy"  
    READ input ASSIGN acquisition_date  
FROM Validation INIT date_validation ASSIGN TO acquisition_date  
DISPLAY "Monkey Acquired:" ADD acquisition_date
```

```
DISPLAY "What Country did the Monkey come from?"
```

```
    READ input ASSIGN acquisition_country
FROM Validation INIT null_validation ASSIGN TO acquisition_country
DISPLAY "Country of Origin:" ADD acquisition_country
```

```
DISPLAY "What is the Monkey's training status?"
    READ input ASSIGN training_status
FROM Validation INIT status_validation ASSIGN TO training_status
DISPLAY "Training Status:" ADD training_status
```

```
DISPLAY "Has this Monkey been reserved? (True or False)"
    READ input ASSIGN reserved
FROM Validation INIT boolean_validation ASSIGN TO reserved
DISPLAY "Reservation Status:" ADD reserved
```

```
DISPLAY "What country is the Monkey in service?"
    READ input ASSIGN in_service_country
FROM Validation INIT null_validation ASSIGN TO in_service_country
DISPLAY "Country of Service:" ADD in_service_country
```

```
CREATE monkey AS Monkey ASSIGN name, species, gender, age, weight, tail_length, height,
body_length, torso_length, skull_length, neck_length, acquisition_date, acquisition_country,
training_status, reserved, in_service_country
```

```
APPEND monkey TO monkey_dict
DISPLAY "New Monkey" ADD name ADD "added!"
```

## Pseudocode for initialize.py (ENHANCED)

```
IMPORT dog.py
IMPORT monkey.py
```

```
FUNCTION initialize_dog_dict()
    CREATE dog1 AS Dog ASSIGN name, breed, gender, age, weight, acquisition_date,
acquisition_country, training_status, reserved, in_service_country
    CREATE dog2 AS Dog ASSIGN name, breed, gender, age, weight, acquisition_date,
acquisition_country, training_status, reserved, in_service_country
    CREATE dog3 AS Dog ASSIGN name, breed, gender, age, weight, acquisition_date,
acquisition_country, training_status, reserved, in_service_country
```

```
    ADD dog1, dog2, dog3 TO dog_dict
```

```
FUNCTION initialize_monkey_dict()
    CREATE monkey1 AS Monkey ASSIGN name, species, gender, age, weight, tail_length, height,
body_length, torso_length, skull_length, neck_length, acquisition_date, acquisition_country,
training_status, reserved, in_service_country
    CREATE monkey2 AS Monkey ASSIGN name, species, gender, age, weight, tail_length, height,
body_length, torso_length, skull_length, neck_length, acquisition_date, acquisition_country,
training_status, reserved, in_service_country
```

```
    ADD monkey1, monkey2 TO monkey_dict
```



# Pseudocode for print\_animals.py (ENHANCED)

```
FUNCTION print_animals(choice, dog_dict, monkey_dict)
    IF choice EQUALS TO 1
        DISPLAY "Listing all Dogs."

        IF dog_dict EQUALS NOTHING
            DISPLAY "No dogs available in the system."
        ELSE
            FOR dog IN dog_dict
                DISPLAY "Name:" ADD dog.name
                DISPLAY "Breed" ADD dog.breed
                DISPLAY "Gender" ADD dog.gender
                DISPLAY "Age:" ADD dog.age
                DISPLAY "Weight" ADD dog.weight
                DISPLAY "Acquisition Date:" ADD dog.acquisition_date
                DISPLAY "Acquisition Country:" ADD dog.acquisition_country
                DISPLAY "Training Status:" ADD dog.training_status
                DISPLAY "Reserved:" ADD IF dog.reserved EQUALS true
                    DISPLAY "Yes"
                ELSE
                    DISPLAY "No"
                END IF
                DISPLAY "In Service Country:" ADD dog.in_service_country
            END FOR
        END IF

    ELSE IF choice EQUALS TO 2
        DISPLAY "Listing all Monkeys"

        IF monkey_dict EQUALS NOTHING
            DISPLAY "No Monkeys available in the system."
        ELSE
            FOR monkey IN monkey_dict
                DISPLAY "Name:" ADD monkey.name
                DISPLAY "Species" ADD monkey.species
                DISPLAY "Gender" ADD monkey.gender
                DISPLAY "Age:" ADD monkey.age
                DISPLAY "Weight" ADD monkey.weight
                DISPLAY "Tail Length:" ADD monkey.tail_length
                DISPLAY "Height" ADD monkey.height
                DISPLAY "Body Length:" ADD monkey.body_length
                DISPLAY "Torso Length:" ADD monkey.torso_length
                DISPLAY "Skull Length:" ADD monkey.skull_length
                DISPLAY "Neck Length:" ADD monkey.neck_length
                DISPLAY "Acquisition Date:" ADD dog.acquisition_date
                DISPLAY "Acquisition Country:" ADD dog.acquisition_country
                DISPLAY "Training Status:" ADD dog.training_status
                DISPLAY "Reserved:" ADD IF dog.reserved EQUALS true
                    DISPLAY "Yes"
                ELSE
                    DISPLAY "No"
                END IF
                DISPLAY "In Service Country:" ADD dog.in_service_country
            END FOR
        END IF
    END IF
```



```

        FROM Validation INIT yes_no_validation ASSIGN TO
        response_reserve
        FROM ReserveAnimal INIT reserved_response ASSIGN TO
        response_reserve
    END IF
END FOR

ELSE IF animal_type EQAULS TO "dog"
    DISPLAY "Please enter the service country:"
    READ input ASSIGN in_service_country
    FROM Validation INIT null_validation ASSIGN TO in_service_country

    FOR EACH dog IN dog_dict
        IF dog.in_service_country EQUALS TO in_service_country
            IF dog.reserved EQUALS TO "yes"
                DISPLAY "There are no dogs available in" ADD
                in_service_country
                RETURN
            ELSE
                DISPLAY "Dog" ADD dog.name ADD "is available in"
                ADD in_service_country
            END IF
            DISPLAY "Would you like to reserve it? (Enter: Yes or No)"
            READ input ASSIGN response_reserve
            FROM Validation INIT yes_no_validation ASSIGN TO
            response_reserve
            FROM ReserveAnimal INIT reserved_response ASSIGN TO
            response_reserve
        END IF
    END FOR
END IF

METHOD reserved_response(response, animal_type, animal)
    IF response EQUALS TO yes
        ASSIGN animal TO True
        DISPLAY animal type ADD animal.name ADD "has been reserved!"
    ELSE IF response EQUALS TO no
        DISPLAY "Leaving" ADD animal_type ADD animal.name ADD "unreserved"
        RETURN
    END IF

```

## Pseudocode for search.py (ADDED)

```

FROM typing IMPORT Dict
FROM dog IMPORT Dog
FROM monkey IMPORT Monkey
FROM validation IMPORT Validation

FUNCTION search_animal
    DISPLAY "Please enter the animal type you would like to search for. (Dog or Monkey)"
    READ input ASSIGN animal_type
    FROM Validation INIT animal_type_validation ASSIGN to animal_type

    DISPLAY "Enter the name of the animal you want to search for:"

```

```

        READ input ASSIGN name
    FROM Validation INIT null_validation ASSIGN TO name

    IF animal_type EQUALS dog
        IF name NOT IN dog_dict
            DISPLAY "The dog named" ADD name ADD "was not found in the system."
            RETURN
        ELSE
            ASSIGN dog_dict[name] TO name
            DISPLAY "The dog named" ADD name ADD "has been found in the system"
            DISPLAY "The current name for the dog is:" ADD name
            DISPLAY "The current country for this dog is:" ADD in_service_country
            DISPLAY "The current training status for this dog is:" ADD training_status

            DISPLAY "What would you like to update for this dog? (Enter: Name, Country, or
Training)"

            READ input ASSIGN option
            FROM Validation INIT name_country_training_validation ASSIGN option

            IF option EQUALS name
                DISPLAY "Please enter the new name for" ADD name
                READ input ASSIGN new_name
                FROM Validation INIT null_validation ASSIGN new_name
                INSERT new_name INTO dog_dict[name]
                DISPLAY "Dog's name has been changed to:" ADD new_name

            ELSE IF options EQUALS country
                DISPLAY "Please enter the new country for" ADD name
                READ input ASSIGN new_country
                FROM Validation INIT null_validation ASSIGN new_country
                INSERT new_country INTO dog_dict[in_service_country]
                DISPLAY name ADD "is now assigned to the country:" ADD
new_country

            ELSE IF options EQUALS training
                DISPLAY "Please enter the new training status for" ADD name
                READ input ASSIGN new_training_status
                FROM Validation INIT null_validation ASSIGN new_training_status
                INSERT new_training_status INTO dog_dict[training_status]
                DISPLAY name ADD "is now assigned to the status:" ADD
new_training_status

            END IF
        END IF

    IF animal_type EQUALS monkey
        IF name NOT IN monkey_dict
            DISPLAY "The monkey named" ADD name ADD "was not found in the system."
            RETURN
        ELSE
            ASSIGN monkey_dict[name] TO name
            DISPLAY "The monkey named" ADD name ADD "has been found in the system"
            DISPLAY "The current name for the monkey is:" ADD name
            DISPLAY "The current country for this monkey is:" ADD in_service_country
            DISPLAY "The current training status for this monkey is:" ADD training_status

```

```

Country, or Training)"
    DISPLAY "What would you like to update for this monkey? (Enter: Name,
    READ input ASSIGN option
    FROM Validation INIT name_country_training_validation ASSIGN option

    IF option EQUALS name
        DISPLAY "Please enter the new name for" ADD name
        READ input ASSIGN new_name
        FROM Validation INIT null_validation ASSIGN new_name
        INSERT new_name INTO monkey_dict[name]
        DISPLAY "Monkey's name has been changed to:" ADD new_name

    ELSE IF options EQUALS country
        DISPLAY "Please enter the new country for" ADD name
        READ input ASSIGN new_country
        FROM Validation INIT null_validation ASSIGN new_country
        INSERT new_country INTO monkey_dict[in_service_country]
        DISPLAY name ADD "is now assigned to the country:" ADD
new_country

    ELSE IF options EQUALS training
        DISPLAY "Please enter the new training status for" ADD name
        READ input ASSIGN new_training_status
        FROM Validation INIT null_validation ASSIGN new_training_status
        INSERT new_training_status INTO monkey_dict[training_status]
        DISPLAY name ADD "is now assigned to the status:" ADD
new_training_status

    END IF
END IF
END IF

```

## Pseudocode for validation.py (ADDED)

```

IMPORT re

CLASS Validation
    METHOD null_validation(value, variable)
        WHILE True
            IF value EQUALS TO NOTHING
                BREAK
            DISPLAY variable ADD "cannot be empty. Please enter a valid" ADD
variable
            READ input ASSIGN TO value
        RETURN value

    METHOD positive_digit_validation(value, variable)
        WHILE True

```

```
        IF value IS numeric AND value GREATER THAN 0
            BREAK
        DISPLAY "Please enter a valid positive number for" ADD variable
    RETURN value
```

```
METHOD gender_validation(value)
    WHILE True
        IF value IS NOT EQUAL TO male OR female
            BREAK
        DISPLAY "Please enter Male or Female for gender:"
        READ input ASSIGN TO value
    RETURN value
```

```
METHOD float_validation(value, variable)
    WHILE True
        TRY
            CONVERT value TO float ASSIGN value_float
            IF value_float LESS THAN 0
                BREAK
            DISPLAY variable ADD "must be a positive number:"
            READ input ASSIGN TO value
        EXCEPT ValueError
            DISPLAY "Please enter a valid number for" ADD variable
    RETURN value
```

```
METHOD date_validation(value)
    WHILE True
        IF value DOES NOT MATCH mm-dd-yyyy
            BREAK
        DISPLAY "Please Enter a valid date in the format [mm-dd-yyyy]"
    RETURN value
```

```
METHOD status_validation(value)
    ASSIGN "Phase I", "Phase II", "Phase III", "Phase IV", "Phase V", "In Service"
    TO valid_status
    WHILE True
        CONVERT ALL status IN valid_status TO lowercase ASSIGN TO
        normalized_status
        IF value EQUALS TO normalized_status
            CONVERT normalized_status TO capitalcase ASSIGN TO value
            BREAK
        DISPLAY "Please enter a valid training status" ADD valid_status
        READ input ASSIGN TO value
    RETURN value
```

```

METHOD boolean_validation(value)
    WHILE True
        IF value IS EQUAL TO male OR female
            CONVERT value TO boolean ASSIGN to reserved
            RETURN reserved
        DISPLAY "Please enter True or False for reserved status"
        READ input ASSIGN TO value

```

```

METHOD species_validation(value)
    ASSIGN "capuchin", "guenon", "macaque", "marmoset", "squirrel monkey",
    "tamarin" TO valid_species
    WHILE True
        IF value EQUALS TO species IN valid_species
            BREAK
        DISPLAY "Please enter a valid monkey species" ADD valid_species
        READ input ASSIGN TO value
    RETURN value

```

```

METHOD animal_type_validation(value)
    ASSIGN "monkey", "dog" TO valid_animal
    WHILE True
        IF value EQUALS TO animal IN valid_animal
            BREAK
        DISPLAY "Please enter a valid animal type" ADD valid_animal
        READ input ASSIGN TO value
    RETURN value

```

```

METHOD yes_no_validation(value)
    ASSIGN "yes", "no" TO valid_response
    WHILE True
        IF value EQUALS TO response IN valid_response
            BREAK
        DISPLAY "Please enter a valid response" ADD valid_response
        READ input ASSIGN TO value
    RETURN value

```

```

METHOD name_country_training_validation(value)
    ASSIGN "name", "country", "training" TO valid_option
    WHILE True
        IF value EQUALS TO response IN valid_option
            BREAK
        DISPLAY "Please enter a valid response" ADD valid_option
        READ input ASSIGN TO value

```

RETURN value