**Pet Sanctuary Pseudo Code**

CLASS Pet:

FUNCTION INITIALISE Attributes of Pet Object

(self, sanctuary\_id, animal\_type, breed, vaccinated, neutered, microchip\_num,

admission\_reason, arrival\_date, departure\_date, destination, destination address)

self. sanctuary\_id = sanctuary\_id

self.animal\_type = animal\_type

self.breed = breed

self.vaccinated = vaccinated

self.neutered = neutered

self.microchip\_num = microchip\_num

self.admission\_reason = admission\_reason

self.arrival\_date = arrival\_date

self.departure\_date = departure\_date

self.destination = destination

self.destination\_address = destination\_address

FUNCTION STRING (self)

RETURN STRING = "Sanctuary: " + self.sanctuary\_id

RETURN STRING += "Type: " + self.animal\_type

RETURN STRING += "Breed: " + self.breed

RETURN STRING += "Vaccinated: " + self.vaccinated

RETURN STRING += "Neutered: " + self.neutered

RETURN STRING += "Microchip Number: " + self.microchip\_num

RETURN STRING += "Reason for Admission: " + self.admission\_reason

RETURN STRING += "Date of Arrival: " + self.arrival\_date

RETURN STRING += "Date of Departure: " + self.departure\_date

RETURN STRING += "Destination: " + self.destination

RETURN STRING += "Destination Address: " + self.destination\_address

RETURN STRING

CLASS WildAnimal:

FUNCTION INITIALISE Attributes of WIldAnimal Object

(self, sanctuary\_id, animal\_type, vaccinated, admission\_reason, arrival\_date, departure\_date, destination, destination\_address)

self.sanctuary\_id = sanctuary\_id

self.animal\_type = animal\_type

self.vaccinated = vaccinated

self.admission\_reason = admission\_reason

self.arrival\_date = arrival\_date

self.departure\_date = departure\_date

self.destination = destination

self.destination\_address = destination\_address

FUNCTION STRING (self)

RETURN STRING = "Sanctuary: " + self. sanctuary\_id

RETURN STRING += "Type: " + self.animal\_type

RETURN STRING += "Vaccinated: " + self.vaccinated

RETURN STRING += "Reason for Admission: " + self.admission\_reason

RETURN STRING += "Date of Arrival: " + self.arrival\_date

RETURN STRING += "Date of Departure: " + self.departure\_date

RETURN STRING += "Destination: " + self.destination

RETURN STRING += "Destination Address: " + self.destination\_address

RETURN STRING

Class Treatment:

FUNCTION INITIALISE Attributes of Treatment Object

self, sanctuary\_id, surgery, surgery\_date, medication, med\_start, med\_finish, abused\_by,

abandoned\_by

self.sanctuary\_id = sanctuary\_id

self.surgery = surgery

self.surgery\_date = surgery\_date

self.medication = medication

self.med\_start = med\_start

self.med\_finish = med\_finish

self.abused\_by = abused\_by

self.abandoned\_by = abandoned\_by

FUNCTION STRING (self)

RETURN STRING = "Sanctuary: " + self.sanctuary\_id

RETURN STRING += "Surgery: " + self.surgery

RETURN STRING += "Surgery Date: " + self.surgery\_date

RETURN STRING += "Medication: " + self.medication

RETURN STRING += "Medication Start: " + self.med\_start

RETURN STRING += "Medication Finish: " + self.med\_finish

RETURN STRING += "Responsible for Abuse : " + self.abused\_by

RETURN STRING += "Responsible for Abandoning: " + self.abandoned\_by

RETURN RETURN STRING

**Animal Sanctuary Main**

IMPORT Pet class

IMPORT Wild Animal Class

IMPORT Treatment Class

IMPORT csv

CREATE ARRAY, pet\_list

CREATE ARRAY, wild\_animal\_list

CREATE ARRAY, treatment\_list

CREATE ARRAY, abandoned\_list

CREATE ARRAY, abuser\_list

**Load data from CSV File**

FUNCTION load\_data():

with open CSV file as READ:

Skip first row

FOR row in csv file:

pet\_list. append (PetClass(row[columns]))

waninimal\_list. append (WildAnimalClass(row[columns]))

treatment\_list. append (TreatmentClass(row[columns]))

**Task 2A New Entry**

FUNCTION new\_entry():

OUTPUT Enter Details:

Column Heading Name 1 = INPUT STRING

Coumn Heading Name 2 = INPUT STRING

Coumn Heading Name …\* = INPUT STRING

With open CSV file as append:

writer. writerow ([Name 1, Name 2…\*])

pet\_list. append (PetClass(row[columns]))

waninimal\_list. append (WildAnimalClass(row[columns]))

treatment\_list. append (TreatmentClass(row[columns]))

OUTPUT object

**Task 2B Return Animal Data**

FUNCTION return\_animal\_data ( animal ID ):

User Input “Enter ID: ”

FOR pet in pet\_list:

IF pet.ID == animal ID:

THEN print PET

FOR animal in wanimal\_list:

IF wanimal.ID == animal ID:

THEN print ANIMAL

FOR treatment in treatment\_list:

IF treatment.ID == animal ID:

THEN OUTPUT treatment

**Task 2C Abuse List**

FUNCTION abused():

FOR treatment in treatment\_list:

IF string “name” EXISTS in abused\_by:

THEN append name to abuse\_list

REMOVE duplicate items IN abuse\_list

STORE in temp\_list

INSERTION SORT temp\_list

OUTPUT temp\_list

**Task 2C Abandoned List**

FUNCTION abandoned():

FOR treatment in treatment\_list:

IF string “name” in abandoned \_by:

THEN append name to abandoned\_list

REMOVE duplicate items IN abandoned\_list

STORE in temp\_list AND Sort temp\_list, A to Z

OUTPUT temp\_list

**Task 2E/F Adoption List**

FUNCTION adoption(pet\_type):

FOR pet in pet\_list:

IF pet\_type == animal\_type THEN:

IF pet object has a Microchip Number AND Vaccination == “Yes” AND Neutered == “Yes” AND Admission Reason is (NOT Lost OR NOT Car Accident):

THEN OUTPUT pet object(s)

**Task 2G Return to Owner**

FUNCTION return\_owner\_pet ( ):

temp = []

FOR pet in pet\_list:

IF pet object has a Microchip Number AND Admission Reason == (“lost” OR “car accident”) AND NO departure date:

THEN append pet to temp

INSERTION SORT temp

FOR pet in temp:

OUTPUT pet

FUNCTION return\_owner\_wild ( ):

temp = []

FOR animal in wanimal\_list:

IF Wild Animal object is NOT Abandoned AND NO Departure Date:

THEN append ANIMAL to temp:

INSERTION SORT temp

FOR animal in temp:

OUTPUT animal

**Task 3**

FUNCTION return\_object ( ID ):

FOR object in list:

IF object.sanctuary\_id == ID:

THEN RETURN object

FUNCTION update\_row( object ):

open csv as READ

lines = list(csv.reader)

FOR row in lines:

IF row[0] == object.sanctuary\_id:

THEN row[1] = Column Heading 2

row[2] = Column Heading 3

row[…\*] = Column Heading …\*

open csv as WRITE

wirterows(lines)

OUTPUT (object)

FUNCTION insertion\_sort( list ):

FOR index IN range ( 1, LENGTH of list ):

temp = list [ index ]

i = index – 1

WHILE i >= 0:

IF temp < list[ i ] THEN

list[ i + 1 ] = list[ i ]

list [ i ] = temp

i = i – 1

ElSE THEN:

BREAK

FUNCTION duplicate ( items ):

list = []

FOR item in items:

IF item NOT IN list THEN:

append item to list

RETURN list

FUNCTION return\_object( objects\_ID )

FOR object in list:

IF object.sanctuary\_ID == objects\_ID THEN:

RETURN object

FUNCTION new\_entry\_sub\_menu():

OUTPUT “Pet, Wild Animal or Treatment Data”

OUTPUT “1. Pet Data"

OUTPUT "2. Wild Animal Data"

OUTPUT "3. Treatment Data"

OUTPUT “0. Back to Menu"

INPUT = s

OUTPUT “Enter Selection:”

IF s == 1 THEN:

new\_entry\_pet()

menu()

ELSE IF s == 2 THEN:

new\_entry\_wanimal()

menu()

ELSE IF s == 3 THEN:

new\_entry\_treatment()

menu()

ELSE IF s == 0 THEN:

menu()

ELSE THEN:

OUTPUT “Invalid Input, Please Select Again”

FUNTION update\_attribute\_sub\_menu():

OUTPUT “Enter Sanctuary ID”

INPUT sanctuary\_ID

object = CALL FUNCTION return\_object( INPUT )

OUTPUT “What attribute would you like to change”

OUTPUT “1. Attribute Name 1”

OUTPUT “…\* Attribute Names…\*”

OUTPUT “Enter Selection: ”

s = INPUT

IF s == 1 THEN:

attribute = INPUT “Enter new attribute: ”

object.attribute\_name1 = attribute1

CALL FUNCTION update\_object(object)

IF s == …\* THEN:

object.attribute\_name…\* = attribute…\*

ELSE IF s == 0 THEN:

CALL FUNCTION menu()

ELSE THEN:

OUTPUT “Invalid Input”

FUNCTION update\_entry\_sub\_menu():

OUTPUT “1. Update Pet”

OUTPUT “2. Update Wild Animal"

OUTPUT “3. Update Treatment"

OUTPUT “0. Back to Menu"

s = INPUT, OUTPUT “Enter Selection

IF s == 1 THEN:

CALL FUNCTION update\_pet\_sub\_menu()

ELSE IF s == 2 THEN:

CALL FUNCTION update\_wanimal\_sub\_menu()

ELSE IF s == 3 THEN:

CALL FUNCTION update\_treatment\_sub\_menu()

ELSE IF s == 0:

CALL FUNCTION menu()

ELSE THEN:

OUTPUT “Invalid Input”

FUNCTION adoption\_sub\_menu():

OUTPUT “1. Dog's ready to adopt”

OUTPUT “2. Cat's ready to adopt”

OUTPUT “0. Back to Menu"

OUTPUT "Enter Selection:”

s = INPUT, OUTPUT "Enter Selection:”

IF s == 1 THEN:

CALL FUNCTION adoption('dog')

CALL FUNCTION menu()

ELSE IF s == 2:

adoption('cat')

menu()

elif s == 0:

load\_pet\_data()

menu()

else:

print("Invalid Input")

**Menu**

FUNCTION menu():

OUTPUT “1. Search for Animal”

OUTPUT “2. Create new entry”

OUTPUT “3. Update existing entry”

OUTPUT “4. Abandonment list”

OUTPUT “5. Abuser list”

OUTPUT “6. Adoption list”

OUTPUT “7. Return to owner list”

OUTPUT “8. Quit”

CALL FUNCTION load\_data( )

S = INPUT “Selection”

IF S == 1 THEN

ID = INPUT (“Enter ID”)

CALL FUNCTION return\_animal\_data(ID)

CALL FUNCTION menu ()

ELSE IF S == 2 THEN

CALL FUNCTION new\_entry\_sub\_menu()

ELSE IF S == 3 THEN

CALL FUNCTION update\_entry\_sub\_menu()

ELSE IF S == 4 THEN:

CALL FUNCTION abused()

CALL FUNCTION menu()

ELSE IF S == 5 THEN:

CALL FUNCTION abandoned()

CALL FUNCTION menu()

ELSE IF S = 6 THEN:

CALL FUNCTION adoption\_sub\_menu()

ELSE IF S = 7 THEN:

Return\_owner\_pet()

Return\_owner\_wild()

CALL FUNCTION menu()

ELSE IF S == 8 THEN:

EXIT

ELSE THEN:

OUTPUT “Invalid Input”

CALL FUNCTION menu()

CALL FUNCTION load\_data()

CALL FUNCTION menu()