```
import os, pickle

# user input to build a dictionary that includes a dictionary
# also accessing existing element, modifying, deleting

def open_file():
    try:
        file_object = open('animalsy.pydata', 'rb')
        animalsy = pickle.load(file_object)
        file_object.close()
    except:
        # return an empty dictionary not an array!
       animalsy = {}
    return animalsy

def close_file():
    try:
        file_object = open('animalsy.pydata', 'wb')
        pickle.dump(animalsy, file_object)
        file_object.close()
        print("\nSuccess! Your rando animal list has been saved\n")
    except Exception as e:
        print(e)
        print("\n\tSorry, something went wrong and the list is not saved")

def print_list():
    if len(animalsy) > 0:
        for anim_ty, anim_information in animalsy.items():
            print("\nThe type of animal is %s" % (anim_ty))
            for key in anim_information:
                if key == "name":
                    print("\tThe name is %s" % anim_information[key])
                if key == "color":
                    print("\tThe color is %s" % anim_information[key])
                if key == "temperament":
                    print("\tThe temperament is %s" % anim_information[key])
    else:
        print("There is nothing here yet")


def add_animal():
    anim_ty = input("\nType of animal? ")
    anim_n = "name"
    anim_name = input("Name of animal? ")
    anim_c = "color"
    anim_color = input("Color of animal? ")
    anim_te = "temperament"
    anim_temp = input("What is the temperment? ")

    animalsy.update({anim_ty : {anim_n: anim_name, anim_c : anim_color, anim_te :
anim_temp}})


# if key found then make changes
def change_animal():
    anim_ty = input("What type of animal are you changing? ")
    if anim_ty in animalsy:
        anim_n = "name"
        anim_name = input("Name of animal? ")
        anim_c = "color"
        anim_color = input("Color of animal? ")
        anim_te = "temperament"
        anim_temp = input("What is the temperment? ")
        # change the entry based on the main key of type
        animalsy.update({anim_ty : {anim_n: anim_name, anim_c : anim_color, anim_te :
anim_temp}})
        print("The information for %s is updated" % anim_ty)
    else:
        print("Sorry, that animal does not seem to be here. Check the list again.")

def remove_animal():
    a_type = input("What type of animal do you want to remove? ")
    if a_type in animalsy:
        del animalsy[a_type]
        print("%s was removed " % a_type)
    else:
        print("Sorry, that animal is not here. Check the list again.")

def print_menu():
    os.system('clear')
    print("1 to print the list")
    print("2 to add a new animal")
    print("3 to change an animal")
    print ("4 to remove an animal")
    print("5 to quit")
```

```
# this would be a parent algorithm
# any of the functions called by this could be the children
def process_menu():
    choice = input("\nWhat would you like to do? ")
    print_menu()
    if choice == '1':
        print_list()
    elif choice == "2":
        add_animal()
    elif choice == '3':
        change_animal()
    elif choice == '4':
        remove_animal()
    else:
        if choice != '5':
            print("What are you doing? That is not an option. Try again my friend.")
    return choice

# main
# if file is there open and load else create empty dictionary
animalsy = open_file()

#
choice = ""
print_menu()

# this will keep running until user enters a '5'
while choice != '5':
    choice = process_menu()

# finish up
close_file()
```