Pair Programming 10.1 Turn In

Name: \_\_\_\_\_Matthew Krahel \_\_\_\_\_\_\_\_\_ Username: \_\_\_\_\_\_\_C1010B11\_\_\_\_\_\_\_\_\_

Partner name: \_\_\_\_\_\_\_Blake Hodges\_\_\_\_\_\_\_\_\_\_\_ Partner username: \_\_\_\_C1010B06\_\_\_\_\_\_\_\_

**Things are still going great with Blake**

SCORE: \_\_\_\_\_\_\_\_\_\_\_\_ (to be filled in by instructor)

10a (2 point)

**/\* File: pp10a.cpp**

**\* Author: Blake Hodges and Matthew Krahel.**

**\* This program declares a structure named Pet and stores pet**

**\* data from the user.**

**\*/**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**struct Pet{**

**string name;**

**int age;**

**double weight;**

**bool isNeutered;**

**char boolcheck;**

**};**

**int main(){**

**Pet pet1;**

**cout << "Please Enter your pet's name: " << endl;**

**cin >> pet1.name;**

**cout << "Please Enter " << pet1.name << "'s age in years: " << endl;**

**cin >> pet1.age;**

**cout << "Please Enter " << pet1.name << "'s weight in lbs: " << endl;**

**cin >> pet1.weight;**

**cout << "Is " << pet1.name << " nuetered? (Y or N)" << endl;**

**cin >> pet1.boolcheck;**

**if(pet1.boolcheck == 'Y' || pet1.boolcheck == 'y'){**

**pet1.isNeutered = true;**

**}**

**else{**

**pet1.isNeutered = false;**

**}**

**cout << pet1.name << " is ";**

**if(pet1.age > 1){**

**cout << pet1.age << " years old";**

**}**

**else{**

**cout << pet1.age << " year old";**

**}**

**cout << ", weighs "<< pet1.weight << " lbs, and ";**

**if (pet1.isNeutered){**

**cout << "has been nuetered." << endl;**

**}**

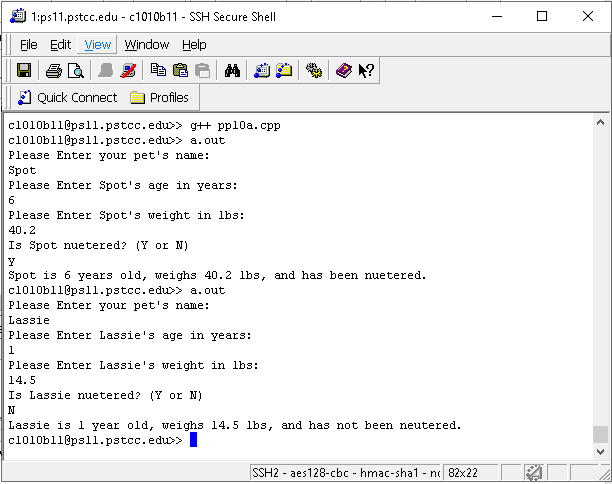
**else{**

**cout << "has not been neutered." << endl;**

**}**

**return(0);**

**}**



10b (2 points)

**/\* File: pp10b.cpp**

**\* Author: Blake Hodges and Matthew Krahel.**

**\* This program declares a structure, calls 2 functions, one to**

**\* fill in the structure and the other to print the info**

**\*/**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**struct Pet{**

**string name;**

**int age;**

**double weight;**

**bool isNeutered;**

**char boolcheck;**

**};**

**//Declare Functions**

**void getPetInfo(Pet& petIn);**

**//Precondition: a parameter that is the Pet Structure.**

**//Postcondition: The structure will be filled with the pet's data**

**void displayPet(const Pet& petOut);**

**//Precondition: a parameter that is the Pet Structure.**

**//Postcondition: output of the Pet Structure's info.**

**int main(){**

**Pet pet1;**

**getPetInfo(pet1); //Call the get pet info function**

**displayPet(pet1); //Call the display pet info function**

**return(0);**

**}**

**void getPetInfo(Pet& petIn){**

**//Precondition: a parameter that is the Pet Structure.**

**//Postcondition: The structure will be filled with the pet's data**

**cout << "Please Enter your pet's name: " << endl;**

**cin >> petIn.name;**

**cout << "Please Enter " << petIn.name << "'s age in years: " << endl;**

**cin >> petIn.age;**

**cout << "Please Enter " << petIn.name << "'s weight in lbs: " << endl;**

**cin >> petIn.weight;**

**cout << "Is " << petIn.name << " nuetered? (Y or N)" << endl;**

**cin >> petIn.boolcheck;**

**if(petIn.boolcheck == 'Y' || petIn.boolcheck == 'y'){**

**petIn.isNeutered = true;**

**}**

**else{**

**petIn.isNeutered = false;**

**}**

**}**

**void displayPet(const Pet& petOut){**

**//Precondition: a parameter that is the Pet Structure.**

**//Postcondition: output of the Pet Structure's info.**

**cout << petOut.name << " is ";**

**if(petOut.age > 1){**

**cout << petOut.age << " years old";**

**}**

**else{**

**cout << petOut.age << " year old";**

**}**

**cout << ", weighs "<< petOut.weight << " lbs, and ";**

**if (petOut.isNeutered){**

**cout << "has been nuetered." << endl;**

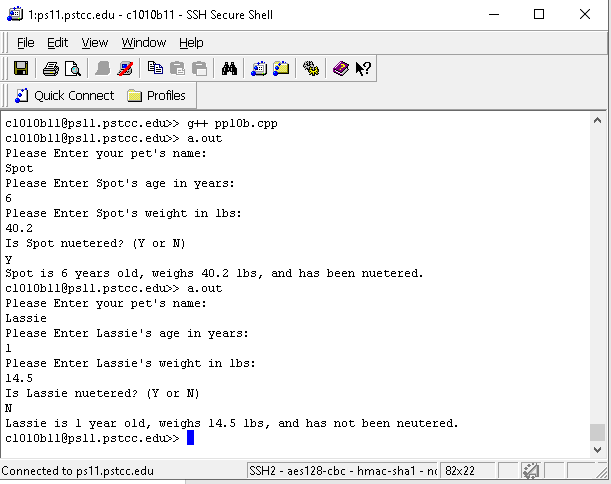
**}**

**else{**

**cout << "has not been neutered." << endl;**

**}**

**}**



10c (3 points)

**/\* File: pp10c.cpp**

**\* Author: Blake Hodges and Matthew Krahel.**

**\* This program declares a structure, builds an array with the structure**

**\* then uses a function to print the array, with column formatting**

**\*/**

**#include <iostream>**

**#include <string>**

**#include <cstdlib>**

**#include <iomanip>**

**using namespace std;**

**struct Pet{**

**string name;**

**int age;**

**double weight;**

**bool isNeutered;**

**};**

**void displayPets(Pet petArr[], int arrSize);**

**//precondition: array with PET structure elements, the max array arrSize**

**//postcondiiton: a formatted pring of the array.**

**main(){**

**//declare const int for array max elements**

**const int ARRAYSIZE = 5;**

**//Assing PET struct array with data.**

**Pet pets[ARRAYSIZE] = {{"Spot", 6, 40.2, true}, {"Lassie", 1, 14.5, false},**

**{"Loner", 1, 8.2, false}, {"Brutus", 11, 92.4, true}, {"Yetti", 16, 28.7, true}};**

**//Call array**

**displayPets(pets, ARRAYSIZE);**

**//return success.**

**return(0);**

**}**

**void displayPets(Pet petArr[], int arrSize){**

**//precondition: array with PET structure elements, the max array arrSize**

**//postcondiiton: a formatted pring of the array.**

**// Table column header formatting**

**cout << setw(10) << left << "Name ";**

**cout << setw(5) << right << "Age ";**

**cout << setw(7) << right << "Weight ";**

**cout << setw(10) << left << "Neutered " << endl;**

**cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**//for loop to print array into strucutred table**

**for(int i = 0; i < arrSize; i++){**

**cout << setw(10) << left << petArr[i].name << " ";**

**cout << setw(5) << right << petArr[i].age << " ";**

**cout << setw(7) << right << petArr[i].weight << " ";**

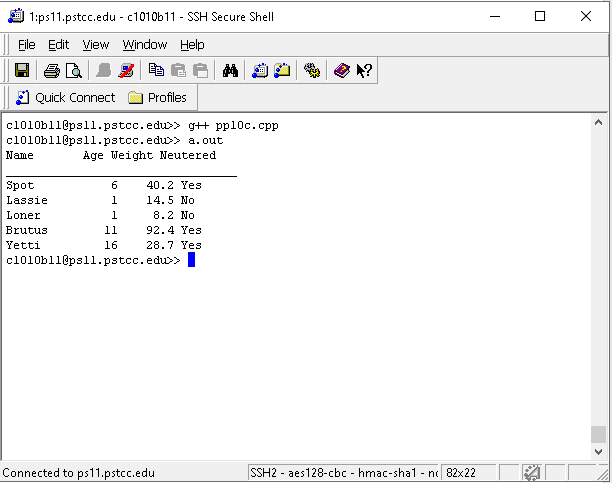
**//boolean element output change**

**if (petArr[i].isNeutered){cout << setw(5) << left << "Yes" << endl;}**

**else {cout << setw(5) << left << "No" << endl;}**

**}**

**}**



10d (3 points)

**/\* File: pp10d.cpp**

**\* Author: Blake Hodges and Matthew Krahel.**

**\* This program declares a structure, builds an array with the structure,**

**\* sorts the array alphabetically then uses a function to print the**

**\* array, with column formatting**

**\*/**

**#include <iostream>**

**#include <string>**

**#include <cstdlib>**

**#include <iomanip>**

**#include<cstring>**

**using namespace std;**

**struct Pet{**

**string name;**

**int age;**

**double weight;**

**bool isNeutered;**

**};**

**void displayPets(Pet petArr[], int arrSize);**

**//precondition: array with PET structure elements, the max array arrSize**

**//postcondiiton: a formatted pring of the array.**

**void sortPets(Pet toSort[], int arraySize);**

**//Precondition: a structure array to be sorted and the number of elements in the array**

**//Postcondition: the array will be sorted based on the name**

**main(){**

**//declare const int for array max elements**

**const int ARRAYSIZE = 5;**

**//Assing PET struct array with data.**

**Pet pets[ARRAYSIZE] = {{"Spot", 6, 40.2, true}, {"Lassie", 1, 14.5, false},**

**{"Loner", 1, 8.2, false}, {"Brutus", 11, 92.4, true}, {"Yetti", 16, 28.7, true}};**

**//Call the sort function**

**sortPets(pets, ARRAYSIZE);**

**//Call array**

**displayPets(pets, ARRAYSIZE);**

**//return success.**

**return(0);**

**}**

**void displayPets(Pet petArr[], int arrSize){**

**//precondition: array with PET structure elements, the max array arrSize**

**//postcondiiton: a formatted pring of the array.**

**// Table column header formatting**

**cout << setw(10) << left << "Name ";**

**cout << setw(5) << right << "Age ";**

**cout << setw(7) << right << "Weight ";**

**cout << setw(10) << left << "Neutered " << endl;**

**cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**//for loop to print array into strucutred table**

**for(int i = 0; i < arrSize; i++){**

**cout << setw(10) << left << petArr[i].name << " ";**

**cout << setw(5) << right << petArr[i].age << " ";**

**cout << setw(7) << right << petArr[i].weight << " ";**

**//boolean element output change**

**if (petArr[i].isNeutered){cout << setw(5) << left << "Yes" << endl;}**

**else {cout << setw(5) << left << "No" << endl;}**

**}**

**}**

**void sortPets(Pet toSort[], int arraySize){**

**//Precondition: a structure array to be sorted and the number of elements in the array**

**//Postcondition: the array will be sorted based on the name**

**string nameTemp[arraySize];**

**int ageTemp[arraySize];**

**double weightTemp[arraySize];**

**bool isNeuteredTemp[arraySize];**

**int alphaTest;**

**//test and sort in alphabetical order**

**for(int i = arraySize-1; i > 0; i--){**

**for(int j = 0; j < i; j++){**

**if(toSort[j].name > toSort[j+1].name){ //true means j string is earlier in the alphabetical**

**nameTemp[0] = toSort[j+1].name;**

**ageTemp[0] = toSort[j+1].age;**

**weightTemp[0] = toSort[j+1].weight;**

**isNeuteredTemp[0] = toSort[j+1].isNeutered;**

**toSort[j+1].name = toSort[j].name;**

**toSort[j+1].age = toSort[j].age;**

**toSort[j+1].weight = toSort[j].weight;**

**toSort[j+1].isNeutered = toSort[j].isNeutered;**

**toSort[j].name = nameTemp[0];**

**toSort[j].age = ageTemp[0];**

**toSort[j].weight = weightTemp[0];**

**toSort[j].isNeutered = isNeuteredTemp[0];**

**}**

**}**

**}**

**}**

