CSC 1300 PROGRAM 2

Fall 2024

Pet Store Management Program

Due Date:

Wednesday, October 9, 2024, by **11:59pm** in the ilearn submission folder named **Program 2**.

You may turn in your submission up to **two days late** with a **penalty of 10 points per day late**. After the two days have passed, the submission folder will **close** in ilearn, and you will not be able to submit. If ilearn marks your assignment as late, then the points will be deducted. **No programs sent outside of the ilearn submission folder will be graded!**

Program Objective

Write a program that simulates a basic pet store management system. The program will allow the user (the shelter manager) to:

- Add animals to the pet store.
- Display a list of animals currently in the pet store.
- Remove an animal from the pet store when it is sold.

You must implement this program using branching, loops, and file input/output. The program will store and retrieve the animal data from a text file.

Important Rules to Remember:

- **Do NOT use ChatGPT, copilot, or any other generative AI** to produce code for this program. This is considered cheating, and you will be charged with academic misconduct and earn a ZERO for this programming assignment.
- **Do NOT work with a partner, friend, or classmate on this program**!! This will be considered cheating, and you will be charged with academic misconduct and earn a ZERO for this programming assignment. Get help from the Teaching Assistants/Mrs. Crockett/Mr. Vandergriff when you get stuck!!
- Include comments throughout your code and a comment block at the top of your source file containing the filename, author, title, and date
- Make your output neat, easy to read, and make sure everything is spelled correctly and uses proper grammar.
- Each programming statement should be on its own line, and you should use consistent indentions.

- We do not recommend that you use functions, since we aren't introducing them until October 7th. However, if you do use functions, your program will be graded making sure good programming practices are followed with functions. So, if we haven't covered that, you may be at risk to get points taken off.
- **Do NOT use programming constructs not yet taught in the class.** This means you **can't use the following in your program**:
    - Arrays / Vectors
    - Pointers
    - Structs
    - Objects/classes
    - Libraries not covered in class (can't use algorithm, map, etc.)
    - Range-based for loops
    - Ternary operators

Specifications:

Name your source file your TTU username, an underscore, and then prog2.cpp. For example, if your TTU email address was jdoe42@tntech.edu, your program 1 source file would be named `jdoe42_prog2.cpp`.

Place your source file in the **CSC 1300 > Programs > Program2** folder on your computer.

Input & Output Files

- The program should read the list of current animals from a text file (**animals.txt**) at the beginning of execution.
- The program should update the list in **animals.txt** as animals are added or removed.
- If the text file isn't in the same folder as the source file, the program will not be able to open it. Make sure you give an error message if the text file can't be opened for input.

```
Error opening file.
```

- The animals.txt file will contain one animal per line in the following format:

```
Dog:Jack
Cat:Loki
Rat:Reuben
Pig:Wilber
Guinea Pig:Chunk
```

Input from the User

- The program should prompt the user for information about the animals they want to add (e.g., animal type and name).
- It should allow the user to remove an animal from the list by specifying its number on the list.

Program Flow

The program should offer the following options in a loop:

```
Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit
```

Your program should validate that the user entered a valid integer. You may assume the user will enter in an integer, but you may not assume they will enter in an integer in the correct range.

Detailed Instructions

Adding Animals:

The program should ask for the animal's type (e.g., "dog" or "cat") and name (e.g., "Jack"). These should be written to the **animals.txt** file, each on a new line. For example, if the user wants to add a Dog named Jack, you would write Dog:Jack to the file.

Displaying Animals:

The program should read from the **animals.txt** file and display the current list of animals along with their assigned number. The format of how you print the animal’s names should be the animal, and then “ named “ and then their name. For example, if you read Dog:Jack from the file, you should print the next incremented number (if it is the first animal, it will be 1) and then Dog named Jack. Refer to sample output for clarification.

Removing Animals:

The program should print the animals to the screen in the same way as when displaying the animals, and then prompt the user for the number of the animal to be removed. This number should be validated to ensure it is a valid number, and then update the **animals.txt** file to reflect the removal.

To do this, you can read from the animals.txt text file and line by line, print all the animals that are not the one to be removed to a temporary file. Then, you can remove the animals.txt text file. Last, you can rename the temporary file to be named animals.txt.

Syntax of removing a file: remove(“name_of_text_file.txt”);

Syntax of renaming a file: rename(“temp.txt”, “animals.txt”); //this renames a file named temp.txt to animals.txt

Grading Assignments - Must Work on Instructor Computer

Programs you submit must work on the instructor’s machine or the grader’s machine to receive full credit. I know this is not awesome but there is not enough time or resources to test your program in a variety of computers/compilers when grading.

If you are worried that you may experience problems, you are welcome to direct message the TAs or your instructor **BEFORE** the day the program is due to have us test your code. We may or may not check email/Teams on Saturday/Sunday. We will test your code one time to make sure it works and return the results to you.

Be aware that MACs are more forgiving than PCs – they will initialize your variables to zero for you and sometimes include files for you. The instructor’s computer is a PC and will not do this for you, which will produce different results.

SAMPLE OUTPUT

User input is highlighted in `yellow`.

```
Welcome to the Pet Store Management System!

Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit

Please choose an option: 2
```

```
Animals currently in the shelter:
1. Dog named Jack
2. Cat named Loki
3. Pig named Wilbur
4. Guinea Pig named Chunk

Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit

Please choose an option: 1

Enter the animal type: Dog
Enter the animal's name: Scout
Scout was added successfully!

Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit

Please choose an option: 2

Animals currently in the pet store:
1. Dog named Jack
2. Cat named Loki
3. Pig named Wilbur
4. Guinea Pig named Chunk
5. Dog named Scout

Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit

Please choose an option:8
Oops! That option doesn’t exist. Please choose 1, 2, 3, or 4.
Please choose an option:3

Animals currently in the pet store:
1. Dog named Jack
2. Cat named Loki
3. Pig named Wilbur
4. Guinea Pig named Chunk
5. Dog named Scout
Enter the number of the animal that sold: 7
Oops! There is no animal with number 7. Select an animal number between 1 and 5.
Enter the number of the animal that sold: 2
The animal has been removed from the file.

Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit
```

```
Please choose an option: 1

Enter the animal type: Plymouth Rock Chicken
Enter the animal's name: Lunch

Choose from the following options:
1. Add a new animal
2. Display all animals
3. Sell an animal
4. Exit

Please choose an option: 2

Animals currently in the shelter:
1. Dog named Jack
2. Pig named Wilbur
3. Guinea Pig named Chunk
4. Dog named Scout
5. Plymouth Rock Chicken named Lunch

Please choose an option: 4

Goodbye!
```

Grade Sheet

Fall 2024 CSC 1300 Program #2
Pet Store Management Program

Earned Points	Possible Points		
	20	**Compiles / Syntax Errors**	
		20	Compiles with no errors – 20 points Does not compile, but only has a very small error like missing one semi-colon – 15 points Does not compile – 0 points
	60	**Follows Program Algorithm & Specifications**	
		10	The menu is printed correctly and continues to repeat until the user selects to Exit (option 4). The menu option is validated with a loop.
		15	Adding an Animal: the program successfully asks the user for the animal and name of the animal and then writes this animal to the animals.txt text file.
		10	Displaying the animals: the program displays the animals in the way specified and shown in sample output by reading from the animals.txt file.
		15	Remove an Animal: the program displays the animals, asks the user for the animal number, validates the user's choice, and then removes the animal from the text file.
		5	Output is spelled correctly, and whitespace is used to make sure output is readable for the user.
		5	A welcome message and goodbye message are printed at the start and end of running the program.
	20	**Readability of Code**	
		5	Comment block at top containing title of program, date, author, and purpose of program.
		5	Sufficient comments in code. Comments should also be spelled correctly.
		5	Code is indented properly (either a tab or 3-4 spaces but must be consistent throughout program).
		5	Variable names are appropriate for the data they are holding, and variables are defined inside the main function (not globally).
	(only subtracts from grade)	**Late Delivery? (10 points per day late)**	
	100	**TOTAL (FINAL) GRADE**	