Programming Exercise 04
Creating a Farm

ISTA-220, C# Step by Step

This activity consists of four programming exercises. The following exercises are open book and open note. You are free to use any written documentation you wish. However, these are individual exercises, and you cannot consult with each other in writing your programs.

This programming exercise has four parts consisting of four requirements. The grade for each requirement is indicated, for a maximum of 100 points. At a minimum, your program must compile successfully and run.

Using classes and objects, implement a farm. Create a class containing a `Main()` function that exercises your other classes. Create classes representing several types of domesticated animals, such as cows, sheep, chickens, goats, etc. Instantiate several objects of each class. For example, for your Rabbit class, you might instantiate objects Bugs Bunny, Roger Rabbit, Thumper, and Peter Cottontail.

Implementing a software test: 70 points In the style of development known as *test driven development* (TDD), the developers write the software test before they write the main application. Ordinarily, developers write the main application first, then they write the testing software. The advantage to TDD is that developers can write the tests based on the initial requirements, and can easily test whether the software meets the requirements.

First, design your farm. Decide on four or more animals. For each animal decide on four or more methods. You should write comments in our software to use as a guide when you implement your program. Using comments to explicitly state your requirements in your source code is a very good practice.

Write a `Main()` method that creates a farm. This will fail because you haven't implemented the classes. It's expected to fail, this is the point. Use exception handling (or other techniques) to ensure that the program runs, even if it does not do anything. The program should emit warnings that particular classes and methods have not been implemented.

One class, one method: 80 points Implement one farm animal by creating a class and a method. For example, you may elect to create a `Horse` type, and a `speak()` method. In your main program, instantiate a `Horse`, name him `MrEd`, and call the `speak()` method. When Mr. Ed speaks, he might say something like, "Hello, my name is Mr. Ed, and I am a horse. I say neigh."

Four classes, one method: 90 points Create four or more animals. Implement a method for each. You can implement a `speak()` method for each animal. A pig can say, "Oink." A cow can say, "Moo." A chicken can say, "Cluck." You get the idea.

Four classes, four methods: 100 points For each class, provide four or more methods. You can have an `eat()` method: horses eat hay, cows eat grass, pigs eat anything, etc. You can have a `product()` method: horses provide transportation, cows provide beef, pigs provide pork, etc. Use your imagination.

This should be a fun project. I hope you enjoy doing this.