



TRAILHEAD
TECHNOLOGY PARTNERS

.NET and C# Training

Session 3

Review Assignment 2

Assignment 2

- Create a console application and use the following base class to define a Cat and Dog class that inherit from it

```
public abstract class Animal
{
    public abstract string Talk();
    public string Feed()
    {
        return Talk();
    }
}
```

- Implement the Talk() method in each class
- In the Main method, create dog and cat instances, feed them, and write out the output.
- Challenge 1: Add a protected field "energy" to the Animal class. Increment it by 1 every time the cat is fed and 2 every time the dog is fed.
- Challenge 2: Add a method to Animal called GetEnergy() that returns the value of the field energy and output the animals new energy every time you feed them

<https://github.com/jonathantower/learning-dotnet>

Today's Agenda

1. Review Assignment 3 (15 min)
2. C# Introduction (3.5 hr)
3. Assignment 3 (15 min)

C# Introduction

Demo

Constructors and Initializers

Demo

Properties

Demo

Private, Public, and Protected

Demo

Methods, Parameters, Output Parameters, and Return Types

Demo

Methods, Parameters, and Return Types

Demo

Output Parameters, Param keyword, Named Parameters, Optional Parameters

Demo

Interfaces

Demo

Class and Interface Inheritance

Demo

Polymorphism

Demo

Abstract, Virtual, Overloading, Shadowing, etc

Demo

Static vs Instance Members

Demo

Assignments

Demo

Constants and ReadOnly

Demo

Loops: while, for, foreach, do...while, break, continue

Demo

Conditionals

Assignment 3

- Write a console application that plays a simple number guessing game
- The game should pick a random number between 1 and 10
- It should then repeatedly prompt the user to guess the number until it is successfully guessed, outputting whether the guess was "too high" or "too low" for wrong guesses
- Sample interaction

```
Guess the number between 1 and 10
5
Too high
3
Too low
4
You win!
```
- Challenge 1: Prompt the user to input the maximum number they want to guess between 1 and X
- Challenge 2: Validate all user input is a valid integer and falls between 1 and X