.Net Training Session 2

1. Visual Studio Walkthrough

2. Basic C# Program

3. Iteration and Flow control

What is object oriented programming?

- the main aspect of object-oriented programming (OOP) is that the program tries to imitate the real world by thinking of things as the objects they represent in the real world.

- So, if you're writing a program for a bank, your classes/objects will be things such as customers, accounts, deposits, withdrawals, checks, loans, etc.

- An easy way to figure out what your objects are going to be is to write down what you want your program to do, and underline all the nouns

An easy way to figure out what your objects are going to be is to write down what you want your program to do, and underline all the nouns. For our game, we want to do these things:

- The player goes to locations.

- The player may need to have certain items to enter a location.

- The location might have a quest available.

- To complete a quest, the player must collect certain items and turn them in.

- The player can collect items by going to a location and fighting monsters there.

- The player fights monsters with weapons.

- The player can use a healing potion while fighting.

- The player receives loot items after defeating a monster.

- After turning in the quest, the player receives reward items.

What's the difference between a class and an object?

- A class is basically a blank form, or blueprint, for an object.

- It defines the object, but it isn't the object.

- Thinking of physical things, a blueprint for a house is a class (it says what the house will be like). Once you have builders follow the blueprint, with wood, steel, concrete, etc., you'll have a house (the object).

- How to create a program (solution) in Visual Studio

- How, and why, to have separate projects in your solution

- Build your First Solution in Visual Studio

- How to determine the properties for a class.

- How to select the correct "datatype" for each property.

- How to create a class.

- Ways to store property values.

- How your program can create objects (instantiation) new ClassName();

- How to assign values to an object's properties

- How to use (reference) classes in different projects

- How to display the values from an object's properties on the UI

Iteration and Flow Control

- If-else

- While

- For Loop

- Foreach

- Switch