MODULE 1: INTRODUCTION TO PROGRAMMING

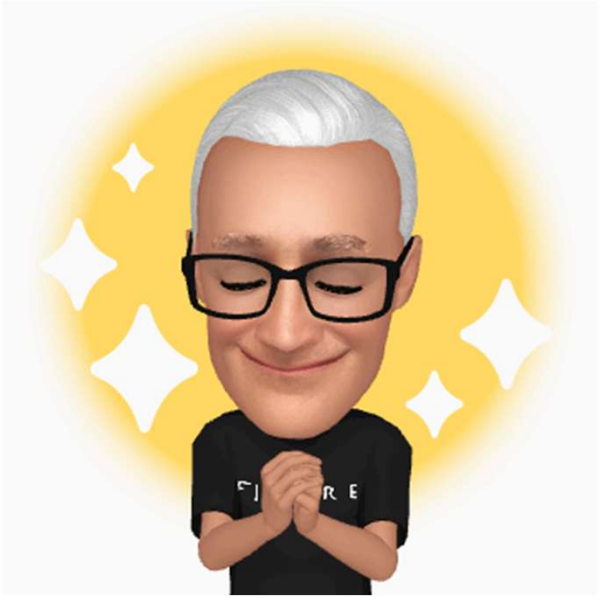
Introduction to Objects Using Strings

Feeling a little….

Last Time

- What is one way to get information from the user?
- How do we give information to the user?

Objects

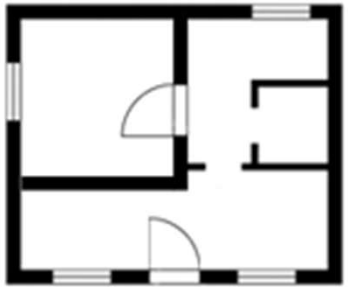


Objects

- An **object** is an in-memory data structure that combines state and behavior into a usable and useful abstraction.

Classes

- A **class** is a grouping of variables and methods in a source code file from which we can generate objects.



Blueprint

Classes

- A **class** is a grouping of variables and methods in a source code file that from which we can generate objects.

Class Object Object

Creating Objects

- First, declare a variable with the type of the object
  - House houseAt901Penn;
- Next, instantiate the new object
  - houseAt901Penn = new House();
- Or, instantiate and initialize the object
  - houseAt901Penn = new House(3,2.5,”Red”);
- All at once
  - House houseAt901Penn = new House(3,2.5,”Red”);

Value Types and Reference Types

- Int
- Bool
- Double
- Float
- Char
- Byte

- Arrays
- Strings
- Objects
- Anything that uses “new”

Our First Object: Strings

- Strings are a special case of an object
- Stored as a collection of chars
- Strings are immutable
  - Example: name.ToUpper() **returns a string**, doesn't change name.

- Initialization doesn't require the "new" keyword
  - string foo = "Hello World";
  - string bar;

Comparing Strings

- How do you see if two ints are equal to each other?
- How do we see if two strings are equal to each other?
- How do we see if two arrays are equal to each other?

Common String Methods

- .Length() : returns the length of a string
- .Substring(): returns part of a string based on the parameters
- .Contains(): returns a bool indicating if the string contains the parameter
- .Startswith(): returns a bool indicating if the string starts with the parameter
- .Endswith(): returns a bool indicating if the string ends with the parameter
- .IndexOf(): returns an int indicating position within the string of the parameter

Common String Methods

- .Replace() : returns new string with characters replaced based on parameters
- .ToLower(): returns string with all the characters lowercase
- .ToUpper(): returns string with all the characters uppercase
- .Equals(): returns a bool indicating if the parameter value equals the string value
- .Split(): returns a string array based on the parameters
- String.Join(): concatenates an array into a string separated by the specified character.

LET'S CODE!

WHAT QUESTIONS DO YOU HAVE?

Reading for tonight:
Collections Part 1