Value and Reference Types

Value Types and Reference Types

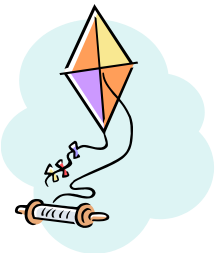
- The data types in C# and the .NET Framework fall into two categories: **values types** and **reference types**
- A variable that is used to hold a value, such as 23, 15.87, “Hello”, etc. is a value type of variable
  - They actually hold data
- A variable that is used to reference an object is commonly called a reference variable
  - Reference variables can be used only to reference objects. They do not hold data.

How a Value Type Works

- When you declare a value type variable, the compiler allocates a chunk of memory that is big enough for the variable
- The memory that is allocated for a value type variable is the actual location that will hold the value assigned to the variable
- When you are working with a value type, you are using a variable that holds a piece of data
- Value type of variable actually holds the data

How a Reference Type Works

- When you work with a reference type, you use two things:
  - An object that is created in memory
  - A variable that references the object
- The object that is created in memory holds data. You need a way to refer to it.
  - A variable is then created to hold a value called **reference**
  - A reference variable does not hold an actual piece of data, it simply refers to the data
  - A reference type links the variable that holds actual data to the object
- If a kite is the object, then the spool of string that holds the site is the reference



Creating a Reference Type

- Two steps are typically required:
  - Declare a reference variable
  - Create an object and associate it with the reference variable
- An example is the **Random** class

```
Random rand = new Random();
```

  - The “Random rand” part declares a variable named “rand”
  - The “new Random()” part creates an object and returns a reference to the object
  - The = operator assigns the reference that was returned from the new operator to the *rand* variable