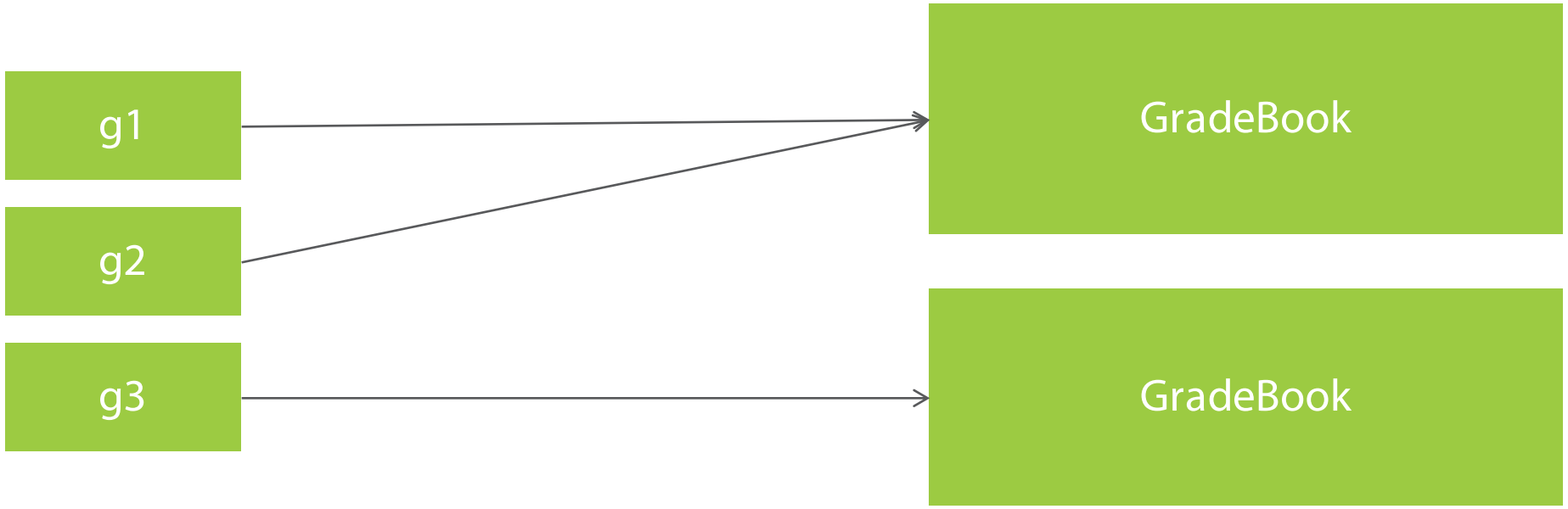
Types

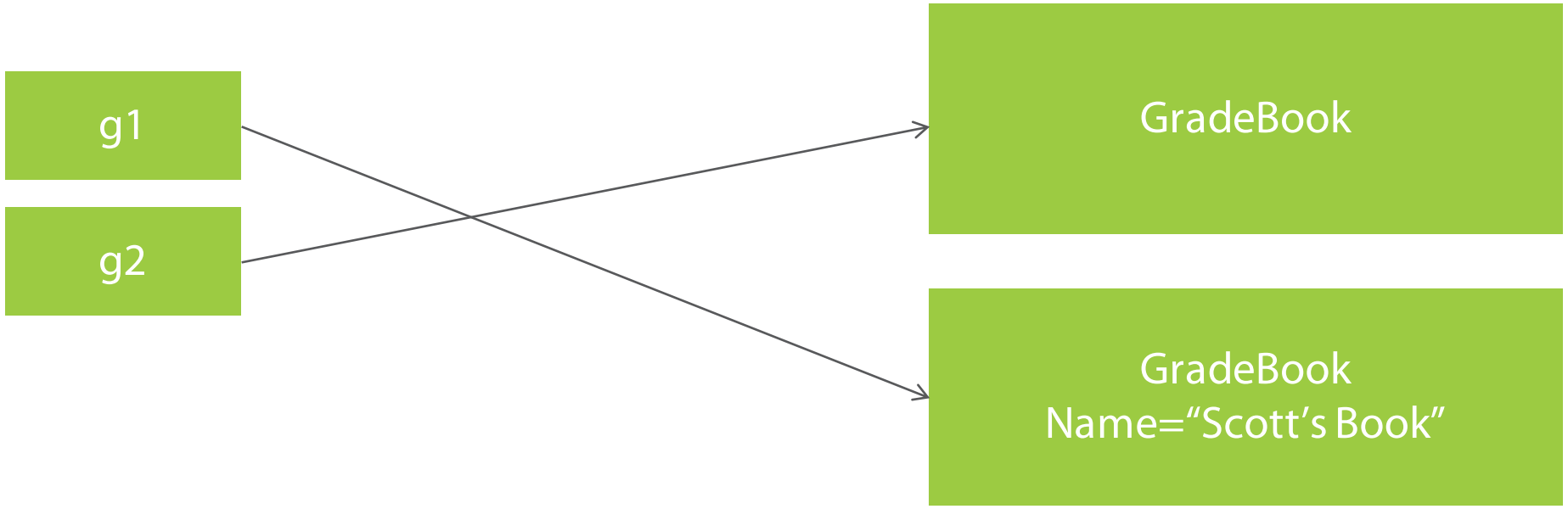
K. Scott Allen

@OdeToCode

Reference Types

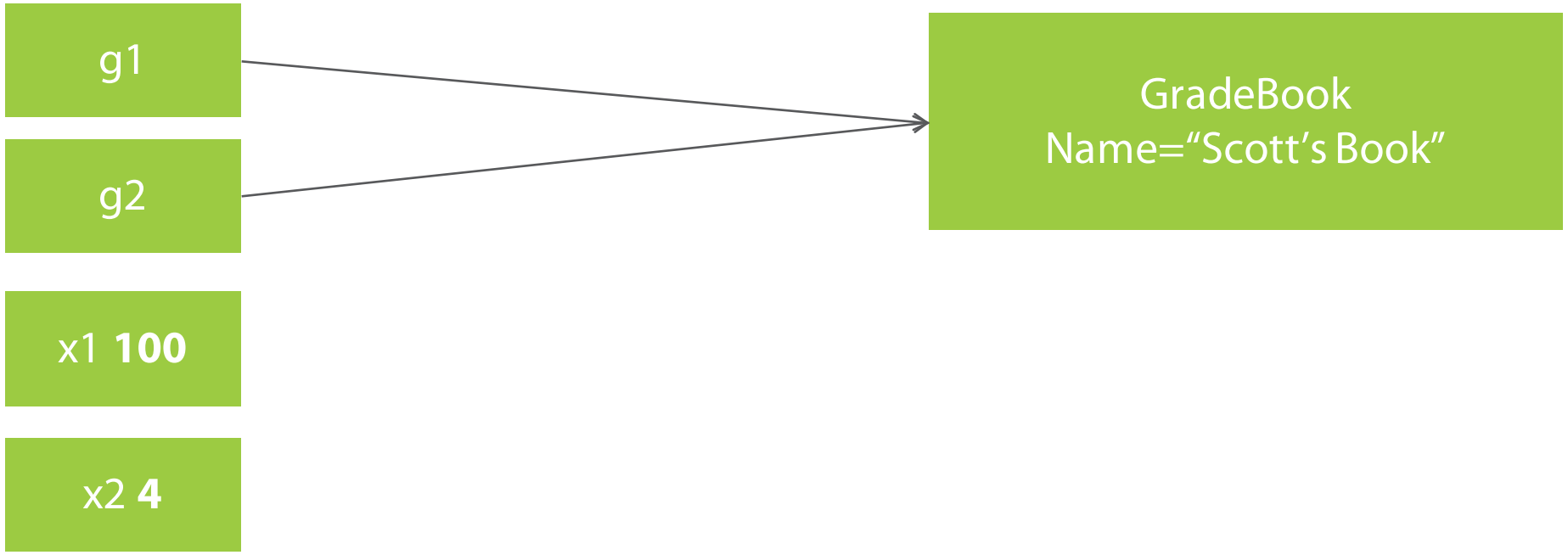
- **Variables store a reference to an object**
  - Multiple variables can point to the same object
  - Single variable can point to multiple objects over it's lifetime
  - Objects allocated into memory using new

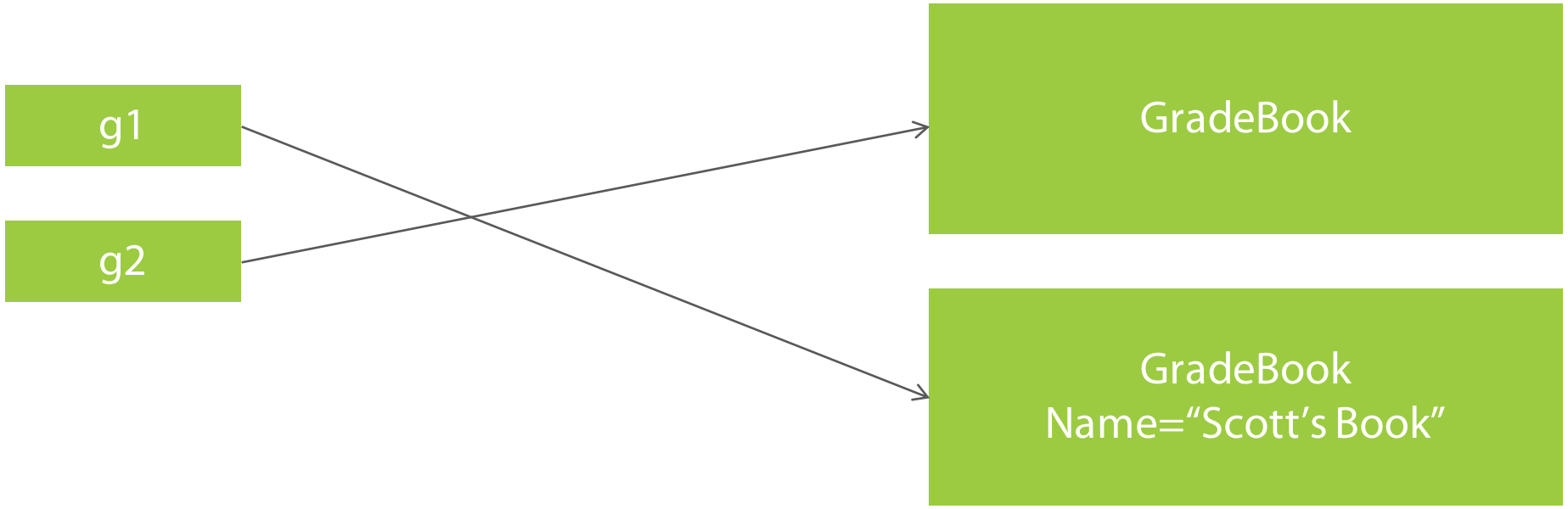


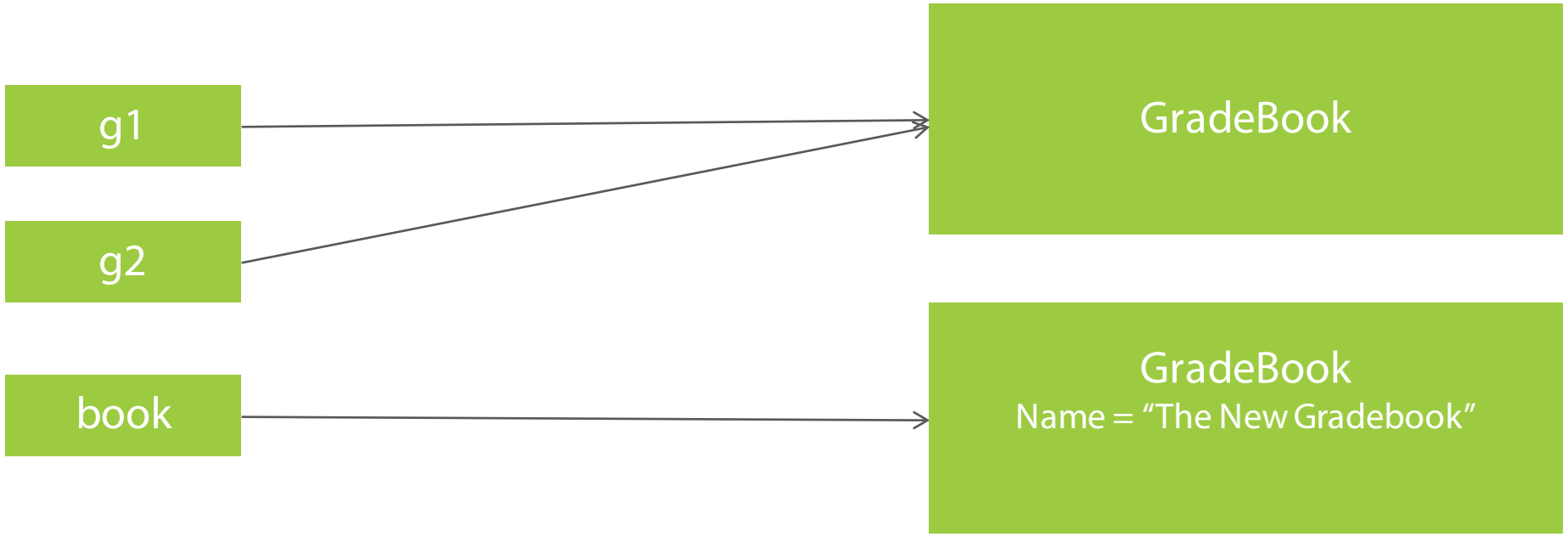


Value Types

- **Variables hold the value**
  - No pointers, no references
- **Many built-in primitives are value types**
  - int, double, float







Creating Value Types

- **struct definitions create value types**
  - Should represent a single value
  - Should be small

```
public struct DateTime
{
    // ...
}
```

Enumerations

- **An enum creates a value type**
  - A set of named constants
  - Underlying data type is int by default

```
public enum PayrollType
{
    Contractor = 1,
    Salaried,
    Executive,
    Hourly
}
```

```
if(employee.Role == PayrollType.Hourly)
{
    // ...
}
```

Method Parameters

- **Parameters pass "by value"**
  - Reference types pass a copy of the reference
  - Value types pass a copy of the value

```
public void DoWork(GradeBook book)
{
    book.Name = "Grades";
}
```

Immutability

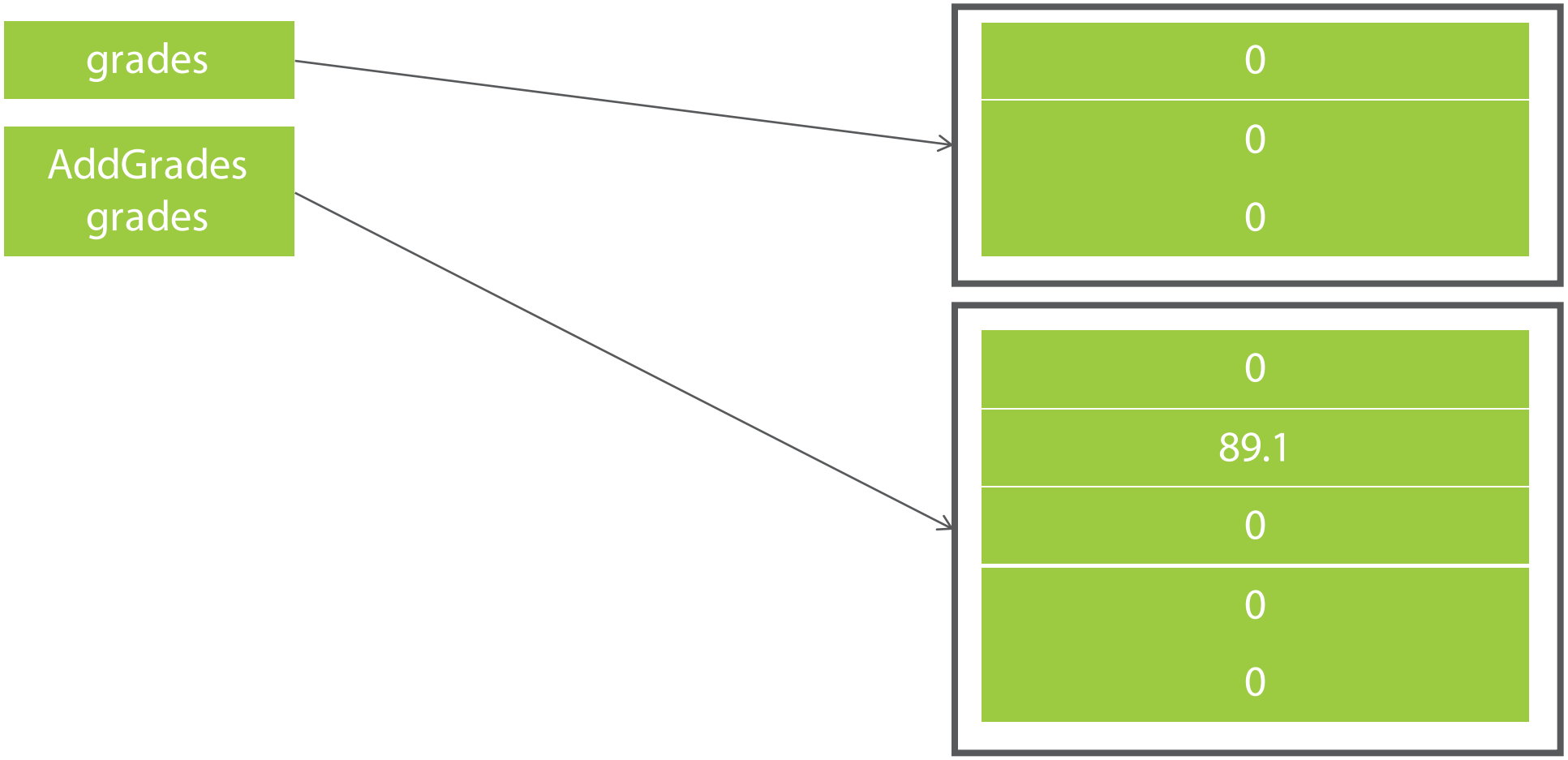
- **Value types are usually immutable**
  - Can not change the value of 4
  - Can not change the value of August 9th, 2002

```
DateTime date = new DateTime(2002, 8, 11);
date.AddDays(1)

string name = " Scott ";
name.Trim();
```

Arrays

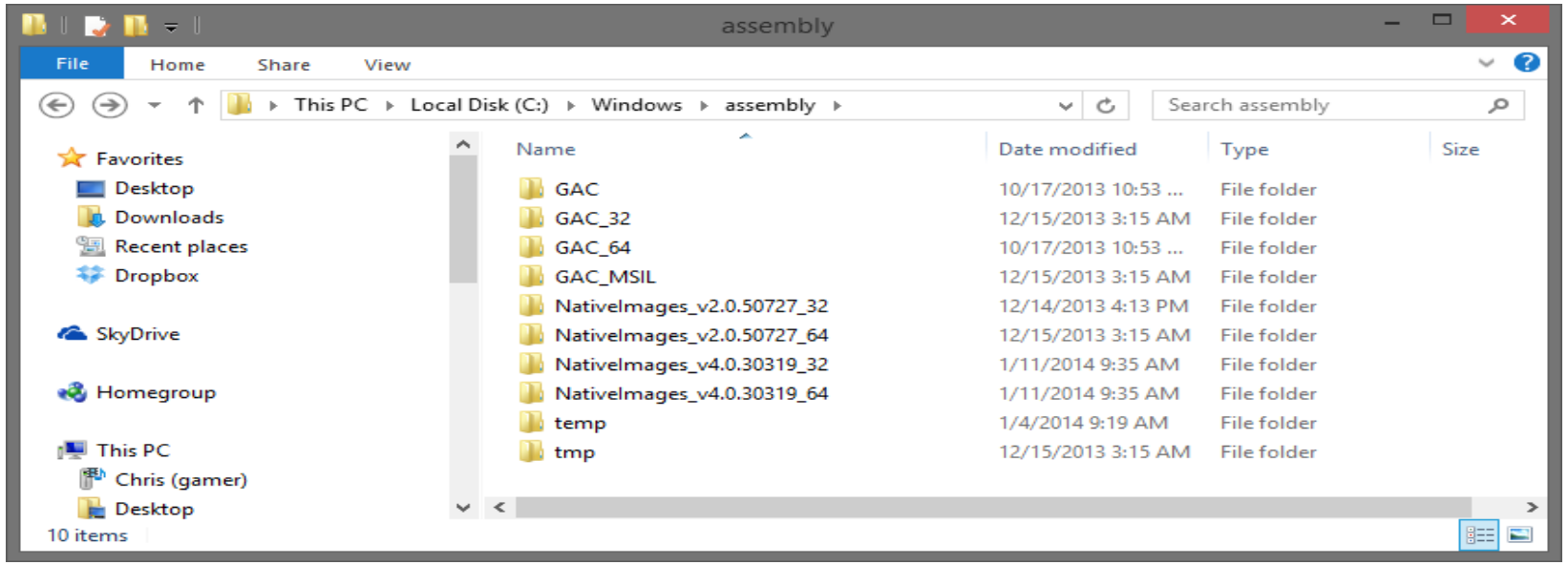
- **Manage a collection of variables**
  - Fixed size
  - Always 0 indexed

```
const int numberOfStudents = 4;
int[] scores = new int[numberOfStudents];

int totalScore = 0;
foreach(int score in scores)
{
    totalScore += score;
}

double averageScore = (double)totalScore / scores.Length;
```



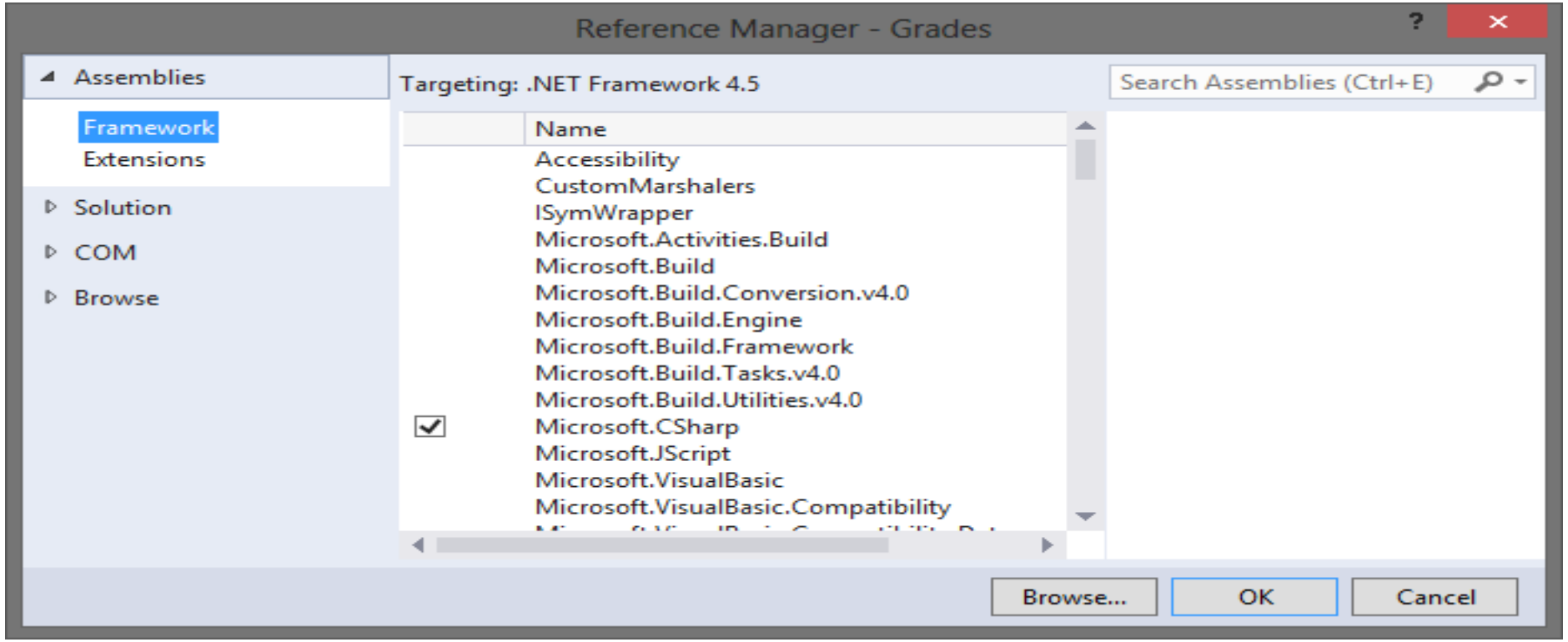
Assemblies

- **Assemblies are .exe or .dll files**
  - Contain metadata about all types inside
- **Global Assembly Cache**
  - A central location to store assemblies for a machine



References

- **Must load assembly into memory before using types inside**
  - Easy approach – reference the assembly in Visual Studio



Summary

- **Every type is a value type or reference type**
  - Use struct to create a value type
  - Use class to create a reference type
- **Arrays and strings are reference types**
  - Strings behave like a value type