



Beginning Level **C#**

Extensible Types

Yong Zhang, Ph.D
Weber State University
2019

Code Reuse: Inheritance + Generics

Inheritance	Generics
Base type	Placeholder type
Is-a relationship	template
Type safety issue	Increase type safety
casting, boxing	Reduce casting and boxing

Why Generics?

- Generics exist to
 - write code that is reusable across **different** types.
 - avoid considerable code duplication.
 - increase type safety
 - reduce casting and boxing

Generics

- Allow code reuse
- Enhance type safety
- Improve performance
- Example: 04-ExtendedTypes/GenericExample

Generic Types

```
public class Stack<T>
{
    int position;
    T[] data = new T[100];
    public void Push (T obj)    => data[position++] = obj;
    public T Pop()              => data[--position];
}
```

```
var stack = new Stack<int>();
stack.Push (5);
stack.Push (10);
int x = stack.Pop();           // x is 10
int y = stack.Pop();           // y is 5
```

Generic Method

```
static void Swap<T> (ref T a, ref T b)
{
    T temp = a;
    a = b;
    b = temp;
}
```

```
int x = 5;
int y = 10;
Swap (ref x, ref y);
```

Introducing Collections

- Example: 04-ExtendedTypes/CollectionExample