Parametrization

Praseed Pai

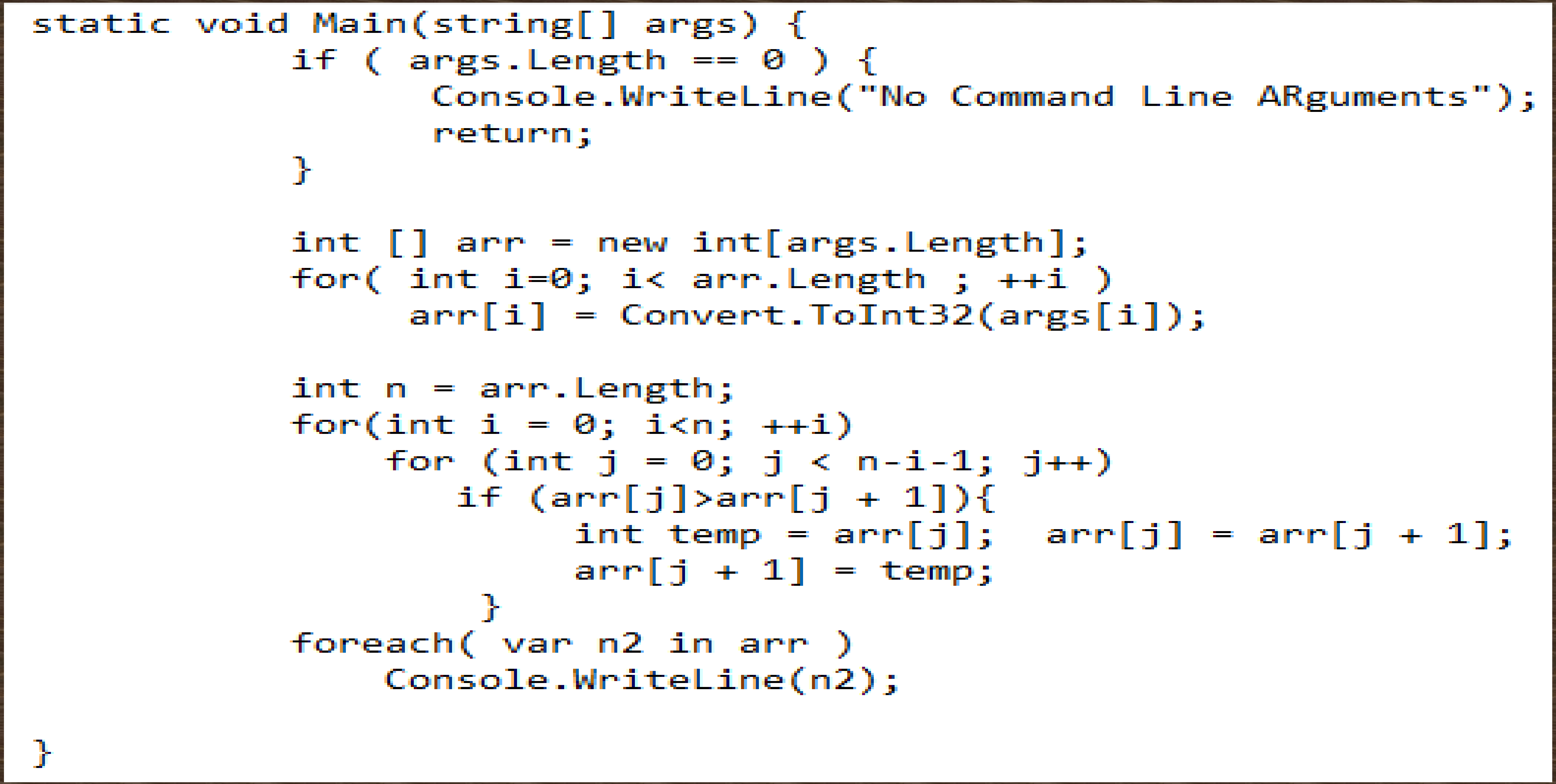
Parametrization Galore!

- Data Parametrization
- Variable Parametrization
- Type Parametrization
- Behavior Parametrization

Data Parametrization

- In our context, we pass data as command line parameter to an application
- The data can be parametrized through OS Text Files/Binary Files
- The data can be parametrized through Storage Solutions (Amazon s3,Azure Blob Storage, Red Hat CEPH)
- A Third option is to use SQL/NoSQL databases
- Another Option available to one is Parametrization through Streams

Example of Data Parametrization in C#

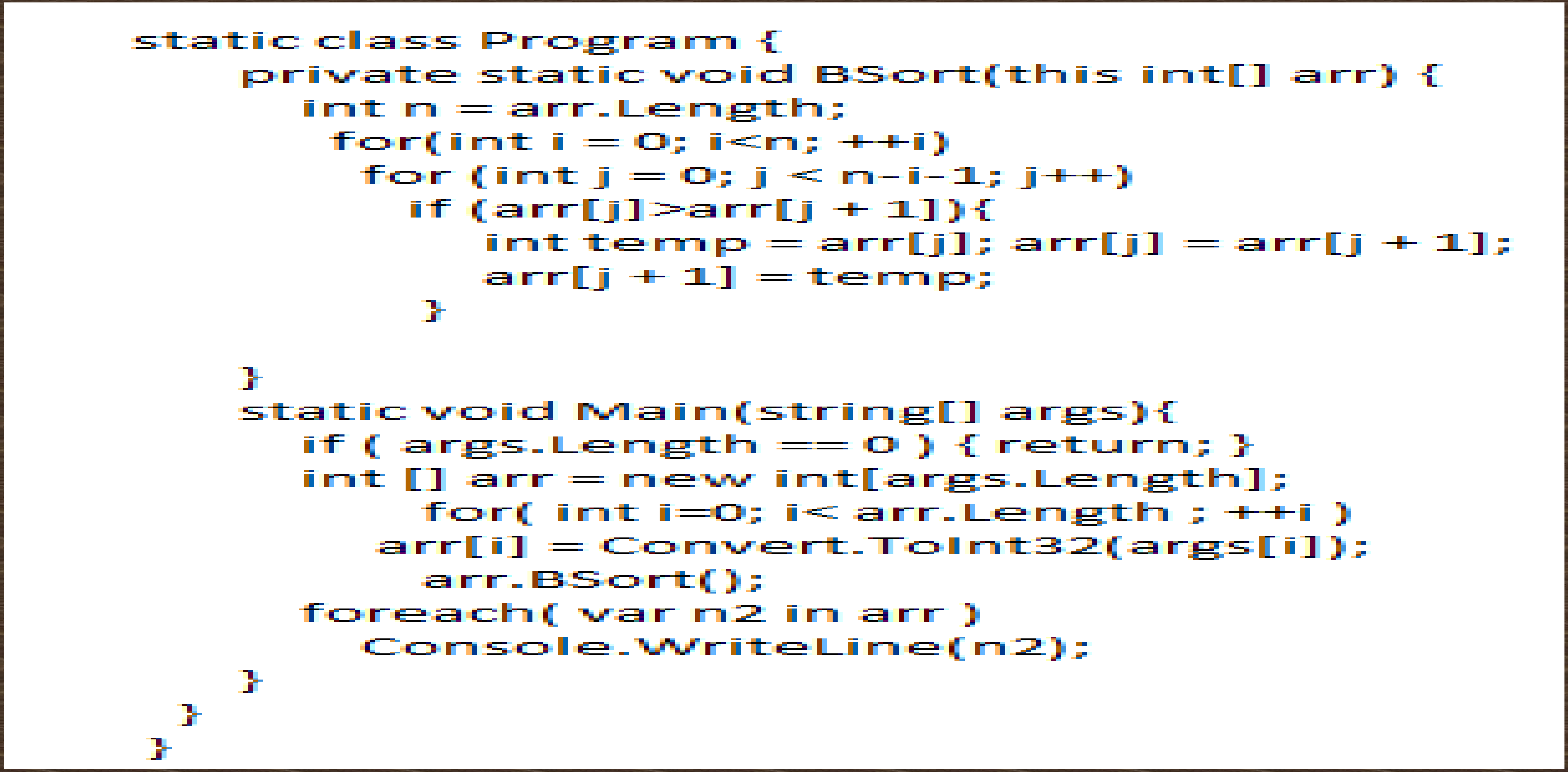


```
static void Main(string[] args) {
           if ( args.Length == 0 ) {
                 Console.WriteLine("No Command Line ARguments");
                 return;
           }

           int [] arr = new int[args.Length];
           for( int i=0; i< arr.Length ; ++i )
                arr[i] = Convert.ToInt32(args[i]);

           int n = arr.Length;
           for(int i = 0; i<n; ++i)
               for (int j = 0; j < n-i-1; j++)
                  if (arr[j]>arr[j + 1]){
                       int temp = arr[j];  arr[j] = arr[j + 1];
                       arr[j + 1] = temp;
                   }
           foreach( var n2 in arr )
               Console.WriteLine(n2);

}
```

Variable Parametrization

- We can pass variables as parameters to Functions/Methods/Procedures
- We can also pass parameters to Lambda/Anonymous Functions/Closure/Blocks

Variable Parametrization

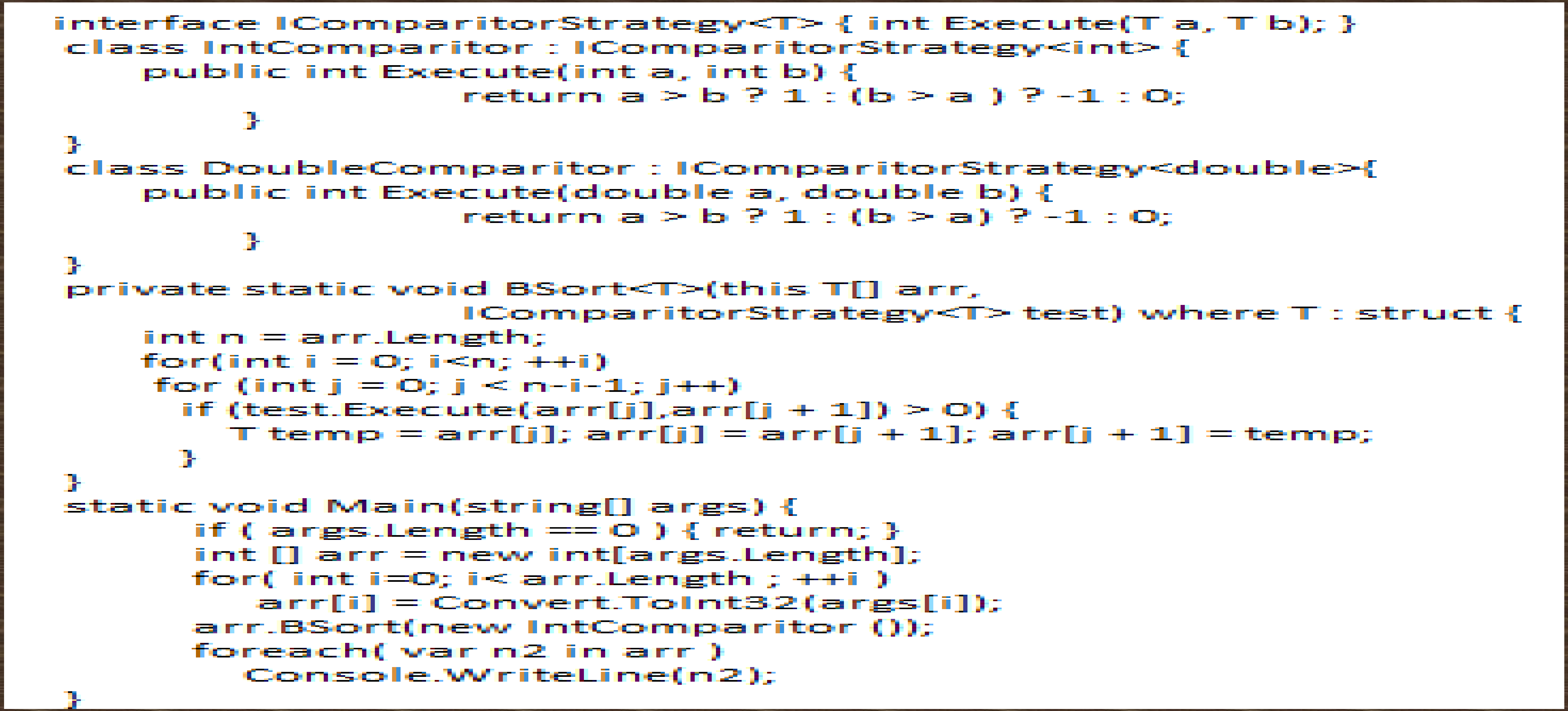


```
static class Program {
    private static void BSort(this int[] arr) {
        int n = arr.Length;
         for(int i = 0; i<n; ++i)
          for (int j = 0; j < n-i-1; j++)
             if (arr[j]>arr[j + 1]){
                 int temp = arr[j]; arr[j] = arr[j + 1];
                 arr[j + 1] = temp;
              }

    }
    static void Main(string[] args){
        if ( args.Length == 0 ) { return; }
        int [] arr = new int[args.Length];
              for( int i=0; i< arr.Length ; ++i )
           arr[i] = Convert.ToInt32(args[i]);
              arr.BSort();
        foreach( var n2 in arr )
          Console.WriteLine(n2);
    }
  }
 }
```

Type Parametrization

- We can implement Parametrized types (Generics) in most modern Programming languages
- Generic Programming (GP) embodies the whole Idea
- Algorithm is the central citizen of GP
- GP is implemented in different ways
  - Compile Time Code factory approach (C++)
  - Type Erasure (Java)
  - Dynamic Type Synthesis (C#)

Type Parametrization in C#

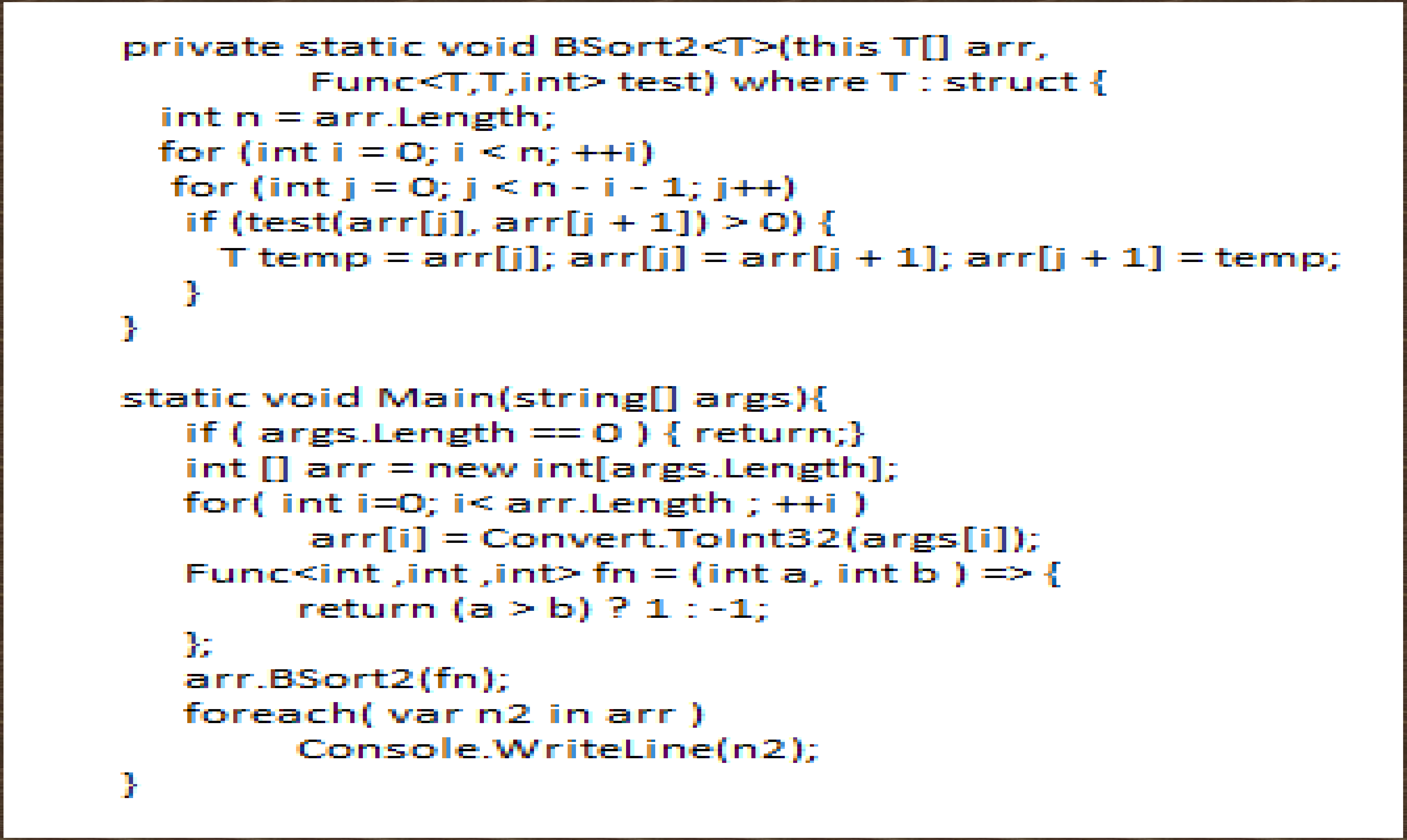


```
interface IComparitorStrategy<T> { int Execute(T a, T b); }
 class IntComparitor : IComparitorStrategy<int> {
      public int Execute(int a, int b) {
                        return a > b ? 1 : (b > a ) ? -1 : 0;
            }
 }
 class DoubleComparitor : IComparitorStrategy<double>{
      public int Execute(double a, double b) {
                        return a > b ? 1 : (b > a) ? -1 : 0;
            }
 }
 private static void BSort<T>(this T[] arr,
                        IComparitorStrategy<T> test) where T : struct {
      int n = arr.Length;
      for(int i = 0; i<n; ++i)
       for (int j = 0; j < n-i-1; j++)
         if (test.Execute(arr[j],arr[j + 1]) > 0) {
            T temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp;
         }
 }
 static void Main(string[] args) {
         if ( args.Length == 0 ) { return; }
         int [] arr = new int[args.Length];
         for( int i=0; i< arr.Length ; ++i )
              arr[i] = Convert.ToInt32(args[i]);
         arr.BSort(new IntComparitor ());
         foreach( var n2 in arr )
            Console.WriteLine(n2);
 }
```

Behavior Parametrization

- Behaviors can be modelled as Lambdas/Blocks/Closures
- Behaviors are also type parametrized

Behavior Parametrization



```
private static void BSort2<T>(this T[] arr,
            Func<T,T,int> test) where T : struct {
   int n = arr.Length;
   for (int i = 0; i < n; ++i)
    for (int j = 0; j < n - i - 1; j++)
     if (test(arr[j], arr[j + 1]) > 0) {
       T temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp;
     }
}

static void Main(string[] args){
     if ( args.Length == 0 ) { return;}
     int [] arr = new int[args.Length];
     for( int i=0; i< arr.Length ; ++i )
             arr[i] = Convert.ToInt32(args[i]);
     Func<int ,int ,int> fn = (int a, int b ) => {
             return (a > b) ? 1 : -1;
     };
     arr.BSort2(fn);
     foreach( var n2 in arr )
             Console.WriteLine(n2);
}
```

Q&A

- If any!
- https://github.com/praseedpai/WhetYourApettite/tree/master/CSharp
- https://github.com/praseedpai/WhetYourApettite/tree/master/JAVA
- https://github.com/praseedpai/WhetYourApettite/tree/master/TypeScript
- https://github.com/praseedpai/WhetYourApettite/tree/master/Python