ClassA.java

```java
package org.eom.oop;

public class ClassA{
    private int value;
      public ClassA(int value){
        this.value = value;
    }
    public int getValue(){
        return value;
    }
    public void setValue(int value){
        this.value = value;
    }
    public void print(){
        System.out.println(value);
    }
}
```

ClassA.cs

```csharp
namespace org.eom.oop{ // not necessarily

    public class ClassA{
        private int value;
        public ClassA(int value){
            this.value = value;
        }
        public int getValue(){
            return value;
        }
        public void setValue(int value){
            this.value = value;
        }
        public void print(){
            System.Console.WriteLine(value);
        }
    }
}
```

GenericClassA.java

```java
package org.eom.oop;

public class GenericClassA<T>{
    private T value;
      public GenericClassA(T value){
        this.value = value;
    }
    public T getValue(){
        return value;
    }
    public void setValue(T value){
        this.value = value;
    }
    public void print(){
        System.out.println(value);
    }
}
```

GenericClassA.cs

```csharp
namespace org.eom.oop{ // not necessarily

    public class GenericClassA<T>{
        private T value;
        public GenericClassA(T value){
            this.value = value;
        }
        public T getValue(){
            return value;
        }
        public void setValue(T value){
            this.value = value;
        }
        public void print(){
            System.Console.WriteLine(value);
        }
    }
}
```

MainClass.java

```java
package org.eom.oop;

public class MainClass {
    public static void main( String[] args ){
        System.out.println( "class and generic class" );

        ClassA objectA = new ClassA(1);
        objectA.print();
        objectA = new ClassA(2);
        objectA.print();

        // GenericClassA<int> intObjectA =
        //     new GenericClassA<int>(1); // not allowed
        // intObjectA.print();
        // intObjectA = new GenericClassA<Integer>(2);
        // intObjectA.print();

        GenericClassA<Integer> integerObjectA =
             new GenericClassA<Integer>(1); // allowed
         integerObjectA.print();
         integerObjectA = new GenericClassA<Integer>(2);
         integerObjectA.print();
    }
}
```

MainClass.cs

```csharp
namespace org.eom.oop{ // not necessarily

    class MainClass{
        static void Main(string[] args){
            System.Console.WriteLine("class and generic class");

            ClassA objectA = new ClassA(1);
            objectA.print();
            objectA = new ClassA(2);
            objectA.print();

            GenericClassA<int> intObjectA =
                new GenericClassA<int>(1); // allowed(int is Int32)
            intObjectA.print();
            intObjectA = new GenericClassA<int>(2);
            intObjectA.print();


            GenericClassA<System.Int32> vInt32ObjectA =
                new GenericClassA<System.Int32>(1); // allowed
            vInt32ObjectA.print();
            vInt32ObjectA = new GenericClassA<System.Int32>(2);
            vInt32ObjectA.print();
        }
    }
}
```

Java		C#	
	`import static java.lang.System.*;`		`using System;`
`System.out.println("example");`	`out.println("example");`	`System.Console.WriteLine("example");`	`Console.WriteLine("example");`
		`System.Int32`	`Int32`

```java
import java.lang.System.*;//all imports
import static java.lang.System.*;//all static fields and methods
```