The Most Difficult Part of the Assignment

The most difficult part of the assignment was adding the arrayReturn method to StringTokenizer. I had small difficulty understanding how to implement this and print the array in the main class.

Status

Completed

Lines of Code

ImprovedRandomizer

```
import java.util.Random;
public class ImprovedRandom extends Random
{
	int firstBoundary;
	int secondBoundary;
	public ImprovedRandom()
	{
		super();
	}

	public ImprovedRandom(long seed)
	{
		super(seed);
	}

	public int getFirstBoundary()
	{
		return firstBoundary;
	}

	public int getSecondBoundary()
	{
		return secondBoundary;
	}

	public void setFirstBoundary(int firstBo)
	{

		firstBoundary=firstBo;
	}

	public void setSecondBoundary( int secondBo)
	{
		secondBoundary=secondBo;
	}

	public int boundaryRandom(int firstB, int secondB)
	{
		firstBoundary=firstB;
		secondBoundary=secondB;
		ImprovedRandom r=new ImprovedRandom();
		return r.nextInt(secondB-firstB+1)+firstB;

	}



}
```

ImprovedStringTokenizer

```
import java.util.StringTokenizer;
public class ImprovedStringTokenizer extends StringTokenizer
{
	String output[];
	int size;
```

```
      ImprovedStringTokenizer(String str)
      {
            super(str);
      }

      ImprovedStringTokenizer(String str, String delim)
      {
            super(str,delim);
      }

      ImprovedStringTokenizer(String str, String delim, boolean returnDelims)
      {
            super(str,delim,returnDelims);
      }

      public int getSize()
      {
            return size;
      }

      public void setSize(int si)
      {
            size=si;
      }

      String [] arrayReturn(String input, String delim)
      {
            ImprovedStringTokenizer stri=new ImprovedStringTokenizer(input,delim);
            size=stri.countTokens();
            output=new String[size];
            while(stri.hasMoreTokens())
            {
                  for(int i=0;i<output.length;i++)
                  {
                        output[i]=stri.nextToken();
                  }
            }
            return output;

      }

}
```

Junit Tests

```
@Test
    public void arrayReturnsTest()
    {
        String input="This class is easy";
        ImprovedStringTokenizer strin=new ImprovedStringTokenizer(input," ");
        String output[]=strin.arrayReturn(input," ");
        assertEquals("This",output[0]);
        assertEquals("class",output[1]);
        assertEquals("is",output[2]);
        assertEquals("easy",output[3]);

    }

    @Test
    public void arrayReturnsTestComma()
    {
        String input="This,class,is,easy";
        ImprovedStringTokenizer strin=new ImprovedStringTokenizer(input,",");
        String output[]=strin.arrayReturn(input,",");
        assertEquals("This",output[0]);
        assertEquals("class",output[1]);
        assertEquals("is",output[2]);
        assertEquals("easy",output[3]);
```

```
    }
```

Cyclomatic Coverage

Both Classes

N=0(No if statements)
P=1 (One Exit Point at the Enf of the Program)
M=E-N+2*P
=1-0+2*P
=3