Class Syntax Primer

Methods:

```
public returnType methodName( declareParameterVariables ){
      //method body
}
```

Constructors:

```
public className(declareParameterVariables ){
      //constructor code
      //initialize instance fields
}
```

Instance Fields:

```
private type variableName;

//Do not initialize the Instance Fields here…Do it in the
//constructor.
```

Local variables:

```
type variableName;              //declaration

       or

type variableName = value; //declaration and initialization

//Variable names must start with a lowercase letter.
//Do not reuse names that have already been used.

//Only declare your variables one time!
```

Example:

```
public class Demo{                  //class header
      private String name;          //declaring an instance field

      public Demo(String input){    //constructor header, with String
                                    //  parameter declared
            name = input;           //assign parameter value to instance field
      }
      public String getName(){      //method header, no parameters,
                                    //String return type
            return name;            //return a String
      }
      public void changeName(String newName){ //method head, String
                                    //paramter, void return type (no return)
            name = newName;         //assign parameter value to instance field
      }
}
```

(More on next page)

```
public class DemoMain(){

	public static void main(String[] args){

		Demo obj = new Demo(“Hopps”);

		String t = obj.getName();

		System.out.println(t);

		obj.changeName(“Mr. Hopps”);

		System.out.println( obj.getName() );
	}
}
```

This DemoMain class creates an object of the Demo type, with an initial name of “Hopps”.

It gets the name out of the object, and prints it.

It changes the name to “Mr. Hopps”.

It prints the name again, this time with “Mr. Hopps” as the output.