

Compiling Classes and a Driver in a Single File

Because of its simplicity, this will be the preferred way of organizing and compiling Java programs in my section of CS 140.

Here is the file Homework.java which contains a class Laugher1 and a driver to test it:

```

////////////////////////////////////
//
// Written by Dr Barry Soroka
//
// Demonstrates combining driver and classes in a single file.
//
////////////////////////////////////
class TestLaugh
{
    public static void main (String [] args)
    {
        Laugher1 x;
        x = new Laugher1();
        x.laugh();
    }

} // end class TestLaugh
////////////////////////////////////
class Laugher1
{
    public Laugher1()
    {
    }

    public void laugh()
    {
        System.out.println("haha");
    }

} // end class Laugher1
////////////////////////////////////

```

Notice how I use a line of slashes (//////////...) to separate classes. In more complex cases, I would put comments at the top of each class.

Similarly, I often separate methods with a row of the form //-----..., and I would comment methods which are not self-explanatory.

Now, let's examine how we compile it.

```

/dfs/user/bisoroka/cs140/bar> ls -l
-rw-----  1 bisoroka csupomona   636 Nov  2 00:00 Homework.java

```

Note that the directory contains only one file, Homework.java.

We now compile it.

```

/dfs/user/bisoroka/cs140/bar> javac Homework.java

```

```

/dfs/user/bisoroka/cs140/bar> ls -l
total 6
-rw-----  1 bisoroka csupomona   636 Nov  2 00:00 Homework.java

```

```
-rw----- 1 bisoroka csupomona 405 Nov 2 00:24 Laugher1.class
-rw----- 1 bisoroka csupomona 319 Nov 2 00:24 TestLaugh.class
```

Notice that we have created *two* class files, one for each class defined inside the file we just compiled (Homework.java).

Notice that Homework is not the name of a class inside the file Homework.java, so we don't a file Homework.class does not get created.

Now let's try running the program. We use the java command and give it the name of the class which contains the method named main. In our case, this is the class TestLaugh.

```
/dfs/user/bisoroka/cs140/bar> java TestLaugh
haha
```

This is the output we expect.

Let's try java Laugher1 to see what happens.

```
/dfs/user/bisoroka/cs140/bar> java Laugher1
Exception in thread "main" java.lang.NoSuchMethodError: main
```

We get an error, because the class Laugher1 does not contain a method named main.

Later in CS 140 or in CS 141, we may try or prescribe other methods for organizing classes and drivers, but, for now, let's put them into a single file.