AP Computer Science with Data Structures

Lab: Java Circus
Part 1

We're now at that point in your Computer Science exploration when we remove your proverbial training wheels kick your bike down the mountain, and chase after you with a shotgun. In this lab, we'll examine the many jagged rocks that line the side of the mountain.

Prelab Exercise: A Java Primer

Read the document entitled "Java Circus Reading" before proceeding to the lab exercises. This document summarizes what we discussed in class and can be used to augment the extensive notes you took.

Make sure that any and all questions you have are answered before going on!

Exercise: My First Java Program

We'll compile and run our first Java program *without using a Java editor or IDE* (Integrated Development Environment). We're going to run an `Acrobat` program that simulates a circus acrobat. Start Notepad. (If you're not using Windows, you'll need to translate this lab for your operating system. Note that you must make a text file, and MAC's don't naturally do this. A MAC can be coerced into creating a text file, but I recommend TextWrangler. You can download it for free.) Type the following code into a new document. Then add appropriate comments.

```
public class Acrobat
{
   private int count;

   public Acrobat()
   {
      count = 0;
   }

   public void clap(int n)
   {
      count = count + n;
   }

   public int count()
   {
      return count;
   }
}
```

Save your code to a file named Acrobat.java in the same directory you'll use for all code in this lab. Let's suppose you chose to save it as c:\java\joy\Acrobat.java.

If you are working on a PC, then before you try compiling this file, it will be helpful to set the "Path" to find Java (if you haven't already done this). Right click on "My Computer" and choose "Properties". Select the Advanced tab and then click Environment Variables. Look for "Path" in the System Variables. Select and click to edit this variable. This process may be slightly different for newer versions of Windows. The folks in Redmond like to keep you on your toes looking for where they have hidden features like this!

The PATH can be a series of directories separated by semicolons. Microsoft Windows looks for programs in the PATH directories in order, from left to right. (You should only have one bin directory for a JDK in the path at a time.) At the right end of the PATH value, add ";C:\Program Files\Java\jdk1.6.0_02\bin" (or appropriate path for wherever you've installed the Java Development Kit), including the semicolon at the beginning. Click "Set", "OK", or "Apply".

From the Start menu, select "Run..." and type "cmd". This will launch a command prompt window showing your current directory. (Mac users should use the terminal window.) Change to the directory where your Acrobat.java file is located. For example, if your source directory is "java\joy" on the C drive, you would type

```
cd c:\java\joy
```

at the prompt, and the prompt should then change to "C:\java\joy". If you type "dir" at the prompt, you should see your file in the directory listing.

Now we'll run "javac.exe" to compile our program from the Acrobat.java text file to an Acrobat.class file containing Java bytecode. At the prompt, type "javac Acrobat.java". If your prompt reappears without error messages, then you have successfully compiled your program. (If you do encounter errors, go fix them, save your file again, and type "javac Acrobat.java" again at the prompt. *Every time you change your code you will need to compile it again before you can run it.*)

The compiler has now generated a Java bytecode file, Acrobat.class. At the prompt, type "dir" to see the new file that was generated. Now type "javap -c Acrobat". This will display the actual sequence of machine code instructions contained in your Acrobat.class file. Cool stuff, huh?

Now we'll run "java.exe" to run the bytecode instructions in Acrobat.class inside the Java Virtual Machine (JVM). Type "java Acrobat" to run your file. You will see the following error message. (This is good. Don't call your teacher over.)

Exception in thread "main" java.lang.NoSuchMethodError: main

The JVM will always try to invoke a special static method called `main` in your class file. (Otherwise, how would it know where in your code to start?) Since you haven't written such a method, you see a *NoSuchMethodError*.

Use Notepad to save the following in a file called AcrobatTest.java in the same directory. Then add appropriate comments.
(Unfortunately, it'll be a while before we understand some of the funny syntax in the `main` method declaration.)

```
public class AcrobatTest
{
   public static void main(String[] args)
   {
      Acrobat alice;
      Acrobat bobbo;
      alice = new Acrobat();
      bobbo = new Acrobat();
      alice.clap(3);
      alice.clap(2);
      System.out.println("Alice:  " + alice.count());
      bobbo.clap(5);
      bobbo.clap(4);
      System.out.println("Bobbo:  " + bobbo.count());
      alice.clap(2);
      System.out.println("Alice:  " + alice.count());
   }
}
```

Compile `AcrobatTest`. Now, when you type "java AcrobatTest", you'll see a listing of Alice and Bobbo's exercise counts. *Make sure these counts are correct.*

Exercise: Genuflection

Add a new method called `kneeBend`, similar to `clap`, that takes an argument indicating how many times your acrobat should perform a knee bend. Be sure to update the same `count` instance variable. Be sure to comment your new method appropriately. Then modify your `AcrobatTest` class to test your new method. Test that your new code runs correctly.

Exercise: I Dub Thee

Modify `Acrobat`'s constructor to take in and store a name for your `Acrobat`. Add a method `getName` to return this name. Modify your `AcrobatTest` class to test this new code.

Show your program to your teacher!