Note that the convention **for Java package names is to use all lowercase letters**, even for **intermediate words**

**……**

Package caveat

It’s worth remembering that anytime you create a package, you implicitly specify a directory structure when you give the package a name. The package *must* live in the directory indicated by its name, which must be a directory that is searchable starting from the CLASSPATH. Experimenting with the **package** keyword can be a bit frustrating at first, because unless you adhere to the package-name to directory-path rule, you’ll get a lot of mysterious runtime messages about not being able to find a particular class, even if that class is sitting there in the same directory. If you get a message like this, try commenting out the **package** statement, and if it runs, you’ll know where the problem lies.

Note that compiled code is often placed in a different directory than source code, but the path to the compiled code must still be found by the JVM using the CLASSPATH.

**--Thinking in Java** 4th Edition “Access Control”

Run Jar file

java -classpath Predit.jar your.package.name.MainClass

or

java –jar yourjarfile.jar (jar file should include Main-Class in Manifest.Mf)