Setting Up Java Packages

Setting the classpath
Packages

- Packages let you organize your code into folders.

- Until you are introduced to packages, the following is not necessary.

- To use packages, we will set the classpath.
Classpath

- Recall the **path** variable that you used to install Java?
  - Told Java where to find its built in classes (like String, Math, Integer, etc.)
    - If your recollection is fuzzy, see my online notes for “Installing Java”.

- The **classpath** is similar
  - Tells Java where to find *your* classes
    - Your classes can live in various folders.
Where Java Expects Packages

By default Java looks for your packages in subdirectories of the current directory

- The current directory is the one that contains your application.
  - i.e., the one from which you are usually compiling

But what if your package lives somewhere else?

- classpath to the rescue
- There are three ways to set the classpath.
Classpath Option 1

- Type the following at the command prompt.
  - `javac myClass.java -classpath path1; path2…`
    - `path1` and `path2` are the directory paths to your packages.
  - Ditto “java” command

Above command will compile `myClass.java` with the correct classpath.

- You will be able to use packages at the specified paths called `path1, path2, etc.`

- e.g., “`javac myClass.java -classpath C:\myCoolCode`”
  - Assumes your packaged classes are in the folder `C:\myCoolCode`
Classpath Option 2

- Type the following at the command prompt.
  - `set classpath=path1; path2...`

- As long as the command prompt is open, the classpath is now set.
  - So type “javac myClass.java” and “java myClass” just like normal.
  - Your packages will be automatically accessed.

- e.g., “set classpath=C:\myCoolCode”
Classpath Option 3

Create an environment variable called `classpath`

- **Advantage**: Will always work for ALL command prompts.
  - won’t have to type the classpath at each prompt
- **Disadvantage**: Will have to be changed if you move your package

How do this?
- Same as `path` variable.
Classpath Option 3 (cont.)

- Read description for getting to path variable.
  - See my “Installing Java” notes.

- Do same to get to classpath.
  - May already exist.
  - If doesn’t exist, then create a new environment variable with the name “classpath”.
The value for the classpath environment variable should be

- ".;C:\myCoolCode"

- C:\myCoolCode is the path to the package
- The "." at the beginning is also a path.
  - It is the current working directory.
  - i.e., it is the default place that Java would have looked for packages.
  - So we include the "." so that we retain the default behavior.
Classpath Option 3 (cont.)

- So if you create a classpath variable with the value ".;C:\myCoolCode"
  - It will look for packaged classes in
    1. "." (the current directory)
    2. "C:\myCoolCode"
Warning

If a classpath environment variable already exists, do not erase the existing values.

- unless you really know what you are doing

Just add your values to the end of the existing list.

- separate your values from the existing values with a semicolon.
Checking Your Classpath

If you are uncertain

- Type “set” at the command prompt.

- Do you see a variable called CLASSPATH = …stuff…?
  - If not, then it is not set.