

# Cmp Sci 187: Programming with Data Structures

## IDE Review

### 1. How are namespace, package, and classpath related?

```
baseDir/x
baseDir/x/y
baseDir/x/y/z
```

```
x.java:
package x.y.z;
public class X {}
```

Where would the java files corresponding to the following classes lie?

The correct location is in the `baseDir/x/y/z` directory which matches with the package declaration for class X.

### 2. What is the Java Archive (aka “jar”) tool?

To Create:

```
jar cf newjarfile.jar path-to-base-dir
```

To Extract:

```
jar xf archivefile.jar
```

To List:

```
jar tf archivefile.jar
```

Any of the above flags can have `v` added, for verbose output:

```
jar xvf archivefile.jar
```

The name and flags are derived from the Unix “tar” (“tape archive”) command. The actual format is essentially the same as zip files (unzip works fine on jar files), plus a “manifest” file that lists the contents of the archive.

### 3. How to Import Assignment Source Files?

#### BlueJ:

Does not directly support importing sources from a jar file but is easy enough to do that. The way to do it is to `unjar` the contents and then ask BlueJ to import from the folder using the **Project->Import** menu item

# Cmp Sci 187: Programming with Data Structures IDE Review

## NetBeans:

Does not directly support importing sources from a jar file but is easy enough to do that. The way to do it is to unjar the contents and then ask NetBeans to create a new project from existing sources. You can also specify the test sources in the same way. NetBeans does not copy these into the project directory but uses them from their current site.

## Eclipse:

You can create a new project using an Archive; that will allow you to import source files from a jar.

## 4. How to submit a Jar?

Submit the updated source jar file that we had handed out, with the **same** name.

## BlueJ:

You can export all your classes in a Jar file. This can be done by  
**Project--> Create Jar File**  
menu item. Make sure that you check the include source box.

## Netbeans:

You can create a jar file containing all the sources. This can be done by editing out the source file exclude filter from the **Properties->Build->Create jar** item in the project properties. The jar produced in Project/dist will then contain the jar files. Alternatively by providing the **Project/src** directory as the base directory to the jar command it can be created manually.

## Eclipse:

You can export all your source classes in a Jar file. This can be done by  
Export -> jar  
Be sure to check to include sources.

# Cmp Sci 187: Programming with Data Structures

## IDE Review

### 5. How to run Tests?

#### BlueJ:

Say you have the jar file somewhere on your file system as `var-packer-assign.jar`. Now you can manually run the tests, just like any other java program by saying (all on one line):

```
java
-classpath junit.jar:var-packer-assign.jar:
    var-packer-test.jar
VarPackerTest.AllTests
```

Note that you need to have **junit.jar**, which is in the **junit.zip** file that can be downloaded from the link on the class resources page. The `var-packer-test.jar` contains the JUnit tests that were supplied with the var-packer assignment.

#### NetBeans:

If we provide you with the source files for JUnit tests you can import them just as the application sources and run them individually by selecting the file and running it, or from the run menu using “run other” item.

If we provide you with just the class files for JUnit tests you will need to modify the `build-impl.xml` file found in the `nbproject` subdirectory of netbeans project folder (can be viewed from the Files panel) and modify the pattern used in detecting the JUnit test classes from

```
default="**/*Test.java"
```

to

```
default="**/*.class"
```

as shown in line 3 of the macro copied below. After that you can run the tests by selecting the project and from the run menu selecting **Run Other->Test Projectname**

```
<target name="-init-macrodef-junit">
<macrodef name="junit" uri="http://www.netbeans.org/ns/j2se-project/2">
  <attribute name="includes" default="**/*.class"/>
```

# **Cmp Sci 187: Programming with Data Structures IDE Review**

## **Eclipse:**

For testing, JUnit class files can be added as External Jar from the

**Project->Properties->Build Path->Library->External Jar**

You also have to add junit.jar as an external library. This can be downloaded from the reference link on the class IDE resources page. To run the test, expand the jar visible as a library in the project and then right click to run as a JUnit test.

## **6. Why did my code throw an exception?**

Look at the stack trace.

## **7. How can I visualize execution?**

Use the debugger to visualize the execution.