

Compiling, Executing, and Jar'ing Java Code

Compiling and Running Simple Java Code:

1. Here is the obligatory "hello world" application:

```
public class T {  
    public static void main(String[] args) {  
        System.out.println("Salut, Le Monde");  
    }  
}
```

which I have placed in `~/Users/yogeshwarkulkarni/Juilee/java/week7_/T.java`.

Compile `T.java` with the `javac` tool:

To run the `T.main()` method, use the `java` interpreter/run-time-compiler tool:

```
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ javac T.java  
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ java T  
Salut, Le Monde
```

2. Now, lets use two Java files; one, `U.java`:

```
public class U {  
    public static void main(String[] args) {  
        Hello.speak();  
    }  
}
```

that refers to a method in the new file, `Hello.java`:

```
public class Hello {  
    public static void speak() {  
        System.out.println("hello");  
    }  
}
```

To compile these two files, use

```
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ javac U.java Hello.java
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ ls
Hello.class    T.class        U.class
Hello.java     T.java         U.java
```

Now you can run **U's** **main()** method via:

```
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ java U
hello
```

CLASSPATH Environment Variable

1. Jump into **/tmp** and now try to run **U**:

```
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ cd /tmp
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ java U
Error: Could not find or load main class U
```

The problem is that the Java interpreter does not know where to find **U.class**. You can specify where to look directly:

```
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ java -classpath /Users/yogeshwarkul
karni/Juilee/java/week7_ T
Salut, Le Monde
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ java -classpath /Users/yogeshwarkul
karni/Juilee/java/week7_ U
hello
```

In general, however, you will not want to repeat the path each time. Java looks for a **CLASSPATH** environment variable and uses it as a colon-separated (on UNIX; PC's use semicolon I think) list of directories. These directories are understood to potentially contain your code. For example, I could set my **CLASSPATH** to include directory **/Users/yogeshwarkulkarni/Juilee/java/week7_**:

```
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ echo $CLASSPATH
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ export CLASSPATH="$CLASSPATH:~/User
s/yogeshwarkulkarni/Juilee/java/week7_
> "
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ echo $CLASSPATH
:~/Users/yogeshwarkulkarni/Juilee/java/week7_
```

now from `/tmp`, you can run your programs without the explicit `java` option:

```
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ cd /tmp
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ java T
Salut, Le Monde
Yogeshwars-MacBook-Pro:tmp yogeshwarkulkarni$ java U
hello
```

Packages and Directory Structure

1. Modify the above code to live in package `foo`:

```
Yogeshwars-MacBook-Pro:~ yogeshwarkulkarni$ cd Juilee/java/week7_
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ mkdir foo
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ cp *.java foo
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ cd foo
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ vi *.java
3 files to edit
```

2. Compile as before (except you are in the `foo`) subdirectory:
3. Now, try to run the simplest class `T`:

```
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ javac *.java
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ java T
Error: Could not find or load main class T
```

4. Now, java is looking for `foo.T` in a file called `foo/T.class`. You are in `foo`, which has no `foo` subdirectory. If you move up a directory, then file `foo/T.class` will be found when you ask for class `foo.T`:

```
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ cd /Users/yogeshwarkulkarni/Juilee/
java/week7_
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ cd foo
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ java foo.T
Salut, Le Monde
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ java foo.T
Salut, Le Monde
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ javac *.java
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ java foo.T
foo: Salut, Le Monde
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ java foo.U
foo: hello
```

Jar'ing Java Code

Without packages:

```
Yogeshwars-MacBook-Pro:foo yogeshwarkulkarni$ cd /Users/yogeshwarkulkarni/Juilee/java/week7_  
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ jar cvf /tmp/tools.jar *.class  
added manifest  
adding: Hello.class(in = 385) (out= 272)(deflated 29%)  
adding: T.class(in = 411) (out= 287)(deflated 30%)  
adding: U.class(in = 283) (out= 221)(deflated 21%)  
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ jar tvf /tmp/tools.jar  
  0 Tue Mar 03 11:12:14 PST 2020 META-INF/  
 69 Tue Mar 03 11:12:14 PST 2020 META-INF/MANIFEST.MF  
385 Tue Mar 03 10:41:54 PST 2020 Hello.class  
411 Tue Mar 03 10:37:58 PST 2020 T.class  
283 Tue Mar 03 10:41:54 PST 2020 U.class
```

With Packages:

```
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ jar cvf /tmp/foo.jar foo  
added manifest  
adding: foo/(in = 0) (out= 0)(stored 0%)  
adding: foo/.Hello.java.swp(in = 12288) (out= 329)(deflated 97%)  
adding: foo/T.class(in = 420) (out= 294)(deflated 30%)  
adding: foo/Hello.class(in = 394) (out= 281)(deflated 28%)  
adding: foo/Hello.java(in = 116) (out= 96)(deflated 17%)  
adding: foo/U.class(in = 291) (out= 227)(deflated 21%)  
adding: foo/U.java(in = 106) (out= 89)(deflated 16%)  
adding: foo/T.java(in = 135) (out= 116)(deflated 14%)  
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ jar tvf /tmp/foo.jar  
  0 Tue Mar 03 11:13:04 PST 2020 META-INF/  
 69 Tue Mar 03 11:13:04 PST 2020 META-INF/MANIFEST.MF  
  0 Tue Mar 03 11:05:18 PST 2020 foo/  
12288 Tue Mar 03 11:01:12 PST 2020 foo/.Hello.java.swp  
 420 Tue Mar 03 11:06:52 PST 2020 foo/T.class  
 394 Tue Mar 03 11:06:52 PST 2020 foo/Hello.class  
 116 Tue Mar 03 11:00:52 PST 2020 foo/Hello.java  
 291 Tue Mar 03 11:06:52 PST 2020 foo/U.class  
 106 Tue Mar 03 11:05:14 PST 2020 foo/U.java  
 135 Tue Mar 03 11:06:40 PST 2020 foo/T.java
```

Executing Java Within a Jar

Without Package:

```
Yogeshwars-MacBook-Pro:week7_ yogeshwarkulkarni$ cd  
Yogeshwars-MacBook-Pro:~ yogeshwarkulkarni$ java -classpath /tmp/tools.jar T  
Salut, Le Monde  
Yogeshwars-MacBook-Pro:~ yogeshwarkulkarni$ java -classpath /tmp/tools.jar U  
hello
```

With Package:

```
Yogeshwars-MacBook-Pro:~ yogeshwarkulkarni$ java -classpath /tmp/foo.jar foo.T  
foo: Salut, Le Monde  
Yogeshwars-MacBook-Pro:~ yogeshwarkulkarni$ java -classpath /tmp/foo.jar foo.U  
foo: hello
```