**Java Basics**

**What are JVM , JRE, JDK and JAR? Can you spell them? :)**

JVM - Java Virtual Machine , same as [pretty much] -

JRE - Java Runtime Environment

JDK - Java Development Kit

JAR - Java Archive

 Screenshots below show classes I’ve created earlier… During the Section on 02/13/2016 I’ve used slightly different package and class names, which are included in the **“Sample”** code folder.

The concepts are the same, so names do not matter ☺

**Java class structure**

achine generated alternative text:
SimpleParser.java B 
+ now); 
1 
2 
3 
4 
6 
8 
9 
11 
12 
13 
14 
16 
17 
18 
19 
20 
21 
package cscie63.very.simple; 
import java. util. Date; 
public class SimpleParser 
public SimpleParser() 
public void print TodaysDate() 
Date now new Date(); 
System. out. print Inc "Time now is: 
args) 
public static void main(StringCJ 
SimpleParser simpleParser 
new SimpleParser(); 
simpleParser . print TodaysDate() ; 

**Java packages and dir structure**

achine generated alternative text:
v CSClE63_VerySimple 
.classpath 
.DS_Store 
.project 
.settings 
bin 
src 
.DS_Store 
cscie63 
.DS_Store 
very 
.DS_Store 
simple 
SimpleParser.java 

package **cscie63.very.simple**;

**What is 'java' and 'javac' ?**

You Java code is in .java files

'**javac**' is a program that 'compiles' the Java code into machine language, byte code - the result of compilation is .class files

Once we have .class files - we can run them by starting the JVM via '**java**' command

Java version:

marinapopova$ java -version

java version "1.8.0\_25"

Java(TM) SE Runtime Environment (build 1.8.0\_25-b17)

Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)

**How to make friends with Classpath and never get "ClassNotFoundException"**

Good reference: <https://www3.ntu.edu.sg/home/ehchua/programming/java/J9c_PackageClasspath.html>

Basic Rules:

To compile:

**javac** -classpath "/dir1/jar1.jar:/dir2/jar2.jar" your/package/yourApp.java

This is where you can get "ClassNotFoundException" !

To run:

This is where you can get "NoClassDefFoundException" !

To run unpackages Java class:

**java** -classpath "/dir1/jar1.jar:/dir2/jar2.jar" your.package.yourApp

To run from a jar if MANIFEST.MF does NOT have Main class:

java -classpath "/dir1/jar1.jar:/dir2/jar2.jar" -jar yourJar.jar your.package.yourApp.java

To run from a jar if MANIFEST.MF does have Main specified:

java -classpath "/dir1/jar1.jar:/dir2/jar2.jar" -jar yourJar.jar

Best practices:

* Do not use system-wide CLASSPATH env variable - always specify your classpath explicitly, like below:

***javac -classpath `hadoop classpath` src/cscie63/section1/aggdates/\*.java -d classes***

* Specify fully qualified path to your java classes:

javac -classpath "/usr/lib/hadoop/hadoop-common-2.5.0-cdh5.3.2.jar:/usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.5.0-mr1-cdh5.3.2.jar" src/edu/hu/bigdata/\*.java -d classes

* Separate .java and .class files:

javac -classpath "/usr/lib/hadoop/hadoop-common-2.5.0-cdh5.3.2.jar:/usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.5.0-mr1-cdh5.3.2.jar" src/edu/hu/bigdata/\*.java -d classes

**Compiling and running our very simple SimpleParser class from the command line:**

I've copied the source code (src dir with all sub-dirs and SimpleParser.java class) from the Eclipse project into a different stand-alone dir: simpleProject/ - to demo that you do not need Eclipse at all to compile/run your Java code:

achine generated alternative text:
simpleProject 
.DS Store 
classes 
.DS Store 
cscie63 
.DS Store 
very 
.DS Store 
simple 
SimpleParser.class 
src 
.DS Store 
cscie63 
.DS Store 
very 
IL .DS_Store 
simple 
SimpleParser.java 

 cd simpleProject/

 mkdir classes

 javac src/cscie63/very/simple/\*.java -d classes

 ls -R classes

java -classpath "./classes" cscie63.very.simple.SimpleParser

Error scenarios:

**Missing classpath**:

marinapopova$ java cscie63.very.simple.SimpleParser

Error: Could not find or load main class cscie63.very.simple.SimpleParser

**Wrong package name**:

marinapopova$ java -classpath "./classes" very.simple.SimpleParser

Error: Could not find or load main class very.simple.SimpleParser