A classpath says: "here is a list of root/top level folders where code may be found."  You don't have to list every single directory that has code... just the root folder(s) the compiler & JVM should start their search from.  All of the subfolders are determined by package & import statements in the class.

For example, imagine that we have code in two different directories

c:\src\com\intertech\transport\Truck.class

c:\libraries\com\intertech\util\VINFormatter.class

The root code folders are: c:\src and c:\libraries

Each class would list the subdirectories they live in, via their package statements (note that the package statement is relative from the root code folder... in other words, you don't write "src" or "libraries" as part of the package statement):

package com.intertech.transport;

public class Truck {

}

and

package com.intertech.util;

public class VINFormatter {

}

Now, let's say we want to compile a class in another root folder:

c:\myapp\com\intertech\inventory\InventoryManager.java

package com.intertech.inventory;

import com.intertech.transport.Truck;

import com.intertech.util.VINFormatter;

public class InventoryManager {

private Truck truck;

private VINFormatter formatter;

// ... more code

public static void main(String[] args) { //... }

}

If we navigated to the c:\myapp directory and just wrote:

javac com\intertech\inventory\InventoryManager.java

or

java com.intertech.inventory.InventoryManager

...the compiler and JVM, respectively, would try to find Truck in c:\myapp\com\intertech\transport and VINFormatter in c:\myapp\com\intertech\util... which are the wrong directories.

So instead, we use the following compile command w/ classpath to let the compiler know it should search for the classes in different root folders:

javac –classpath c:\src;\c:\libraries;. com\intertech\inventory\InventoryManager.java

(the "." means the current directory is also a root folder).

... and we'd run the class with:

java -classpath c:\src;\c:\libraries;. com.intertech.inventory.InventoryManager

This classpath, combined with the package/import statements, are used by the compiler AND the JVM at runtime to find a given class.