THE FIRST LINE OF ANY JAVA FILE THAT YOU OPEN USUALLY LOOKS LIKE THIS

```java
package com.company;

/**
 * Created by vitthal on 23/8/2015.
 */
public interface IShape {

    public double getArea();
    public double getPerimeter();

    default String getColor() {
        return "Color.Red";
    }

}
```

THIS LINE TELLS US WHAT **PACKAGE** THIS PARTICULAR CLASS OR INTERFACE RESIDES IN

THE PACKAGE IS A WAY OF HIERARCHICALLY ORGANIZING FILES INTO **NAMESPACES**

THINK OF A NAMESPACE AS SOMETHING LIKE A DIRECTORY STRUCTURE FOR NAMES OF CLASSES

WITHIN A DIRECTORY, FILE NAMES MUST BE UNIQUE, BUT FILES IN DIFFERENT DIRECTORIES CAN HAVE THE SAME NAME

A FILE CAN BE UNIQUELY IDENTIFIED BY THE COMBINATION OF ITS NAME AND LOCATION

THIS NAME + LOCATION TOGETHER CONSTITUTE THE FILE'S

PATH

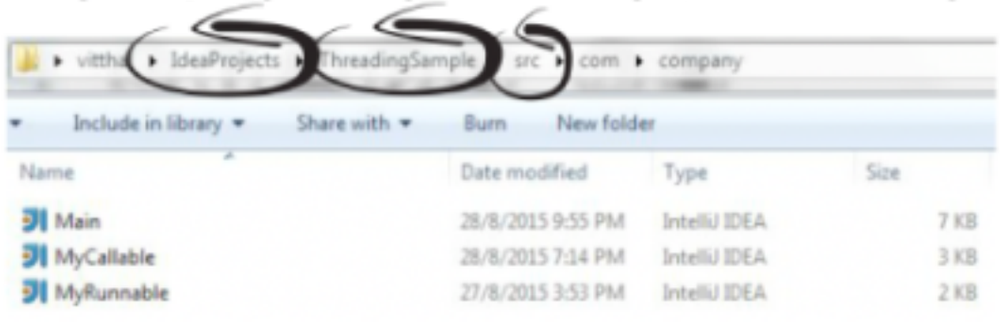
IN FACT THE SIMILARITIES BETWEEN
A PATH AND A PACKAGE/NAMESPACE
ARE NOT COINCIDENTAL -

THE PACKAGE THAT A CLASS BELONGS
TO CORRESPONDS TO THE DIRECTORY
LOCATION OF THE SOURCE CODE (.JAVA)
FILE WITHIN THAT PROJECT

AND IN THE DIRECTORY STRUCTURE OF THE PROJECT

THE FOLDER WHERE THE IDE STORES ALL JAVA PROJECTS

WITHIN THAT FOLDER, A FOLDER FOR THIS PARTICULAR JAVA PROJECT



WITHIN EACH PROJECT FOLDER IS A FOLDER NAMED SRC, WHERE THE CODE RESIDES

INSIDE MAIN.JAVA

```
package com.company;
```

FINALLY, INSIDE THE SRC FOLDER, IS A DIRECTORY HIERARCHY EXACTLY MIRRORING THE PACKAGE NAME

PACKAGES ARE HANDY..ER..PACKAGES..
FOR GROUPING CODE

ZIP THEM ALL INTO A SINGLE JAR FILE
WITH A SINGLE COMMAND

```
jar cf myPackage.jar *.class
```

IMPORT ALL CODE IN A PACKAGE TO MAKE THAT CODE USABLE IN YOUR CODE

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;
```

IN FACT, JAVA BUILT-IN CLASSES ARE MADE AVAILABLE TO PROGRAMMERS USING EXACTLY THIS MECHANISM