

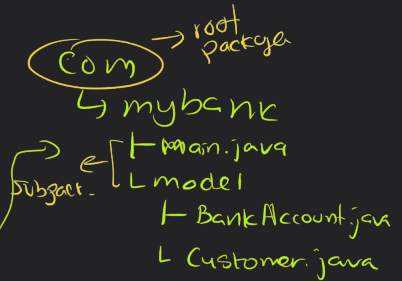
Packages

each dot in the package name denotes a subdirectory

```
package com.mybank.model;  
  
public class BankAccount {  
    //...  
}
```

→ BankAccount & Customer are in ONE package

```
package com.mybank.model;  
  
public class Customer {  
    //...  
}
```



```
package com.mybank;
```

```
import com.mybank.model.BankAccount;
```

→ FQN

```
import com.mybank.model.Customer;
```

→ FQN

```
class Main {
```

→ FQN would be: com.myBank.Main

```
    main(...) {
```

```
        BankAccount account = new BankAccount();
```

```
        Customer customer = new Customer();
```

```
    }  
}
```

→ Default Packages:

- classes are not explicitly placed within any named package.

→ base directory

→ Run vs Compile

• javac HelloWorld.java
vs

java HelloWorld

NOT java HelloWorld.~~class~~

Assumes
current
dir = default
package location

→ let's say current working directory
is packageReview ~~src~~

packageReview (.)

↳ src

↳ HelloWorld.java

→ can't do this: java HelloWorld from pwd

so CLASSPATH

↳ java -cp src HelloWorld

* javac -d bin src/HelloWorld.java

* java -cp bin src/HelloWorld.java
↳
Path

On
java -cp bin HelloWorld

Package CS1302.hello;

→ declare that ①

Ref

Package Renam

└ bin

└ HelloWorld.class

└ src

└ CS1302

└ hello

└ HelloWorld.java

② Place the .java
file in
appropriate pkg dir

Q: N?

CS1302.hello.
HelloWorld.java

→ javac -d bin src/CS1302/hello/HelloWorld.java
destination ← path

→ java -cp bin CS1302.hello.HelloWorld
Package

→ Importing Statements

- `import cs1302.hello.HelloWorld;`

main {
 HelloWorld hello = new HelloWorld();
 HelloWorld.max();
}

→ Max method
in HelloWorld class

OK

OK

- could not import anything, but still use
classes from the package

myPackage.classA objA = new myPackage.classA();
(FQN)

