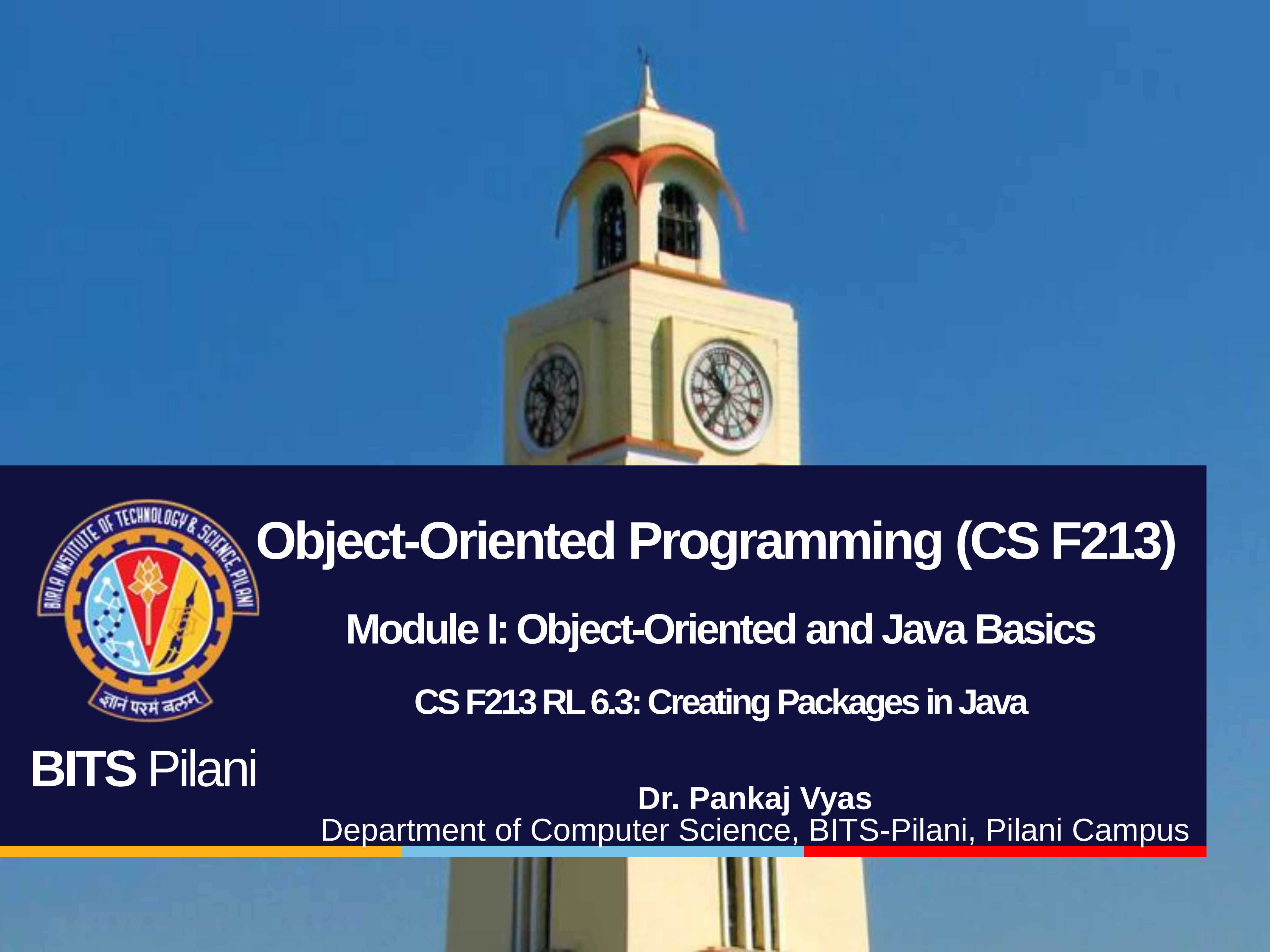


Object-Oriented Programming (CS F213)

Module I: Object-Oriented and Java Basics

CS F213 RL 6.3: Creating Packages in Java

BITS Pilani

Dr. Pankaj Vyas
Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 6.3 : Topics

- Creating Your Own Packages in Java

Packages Introduction

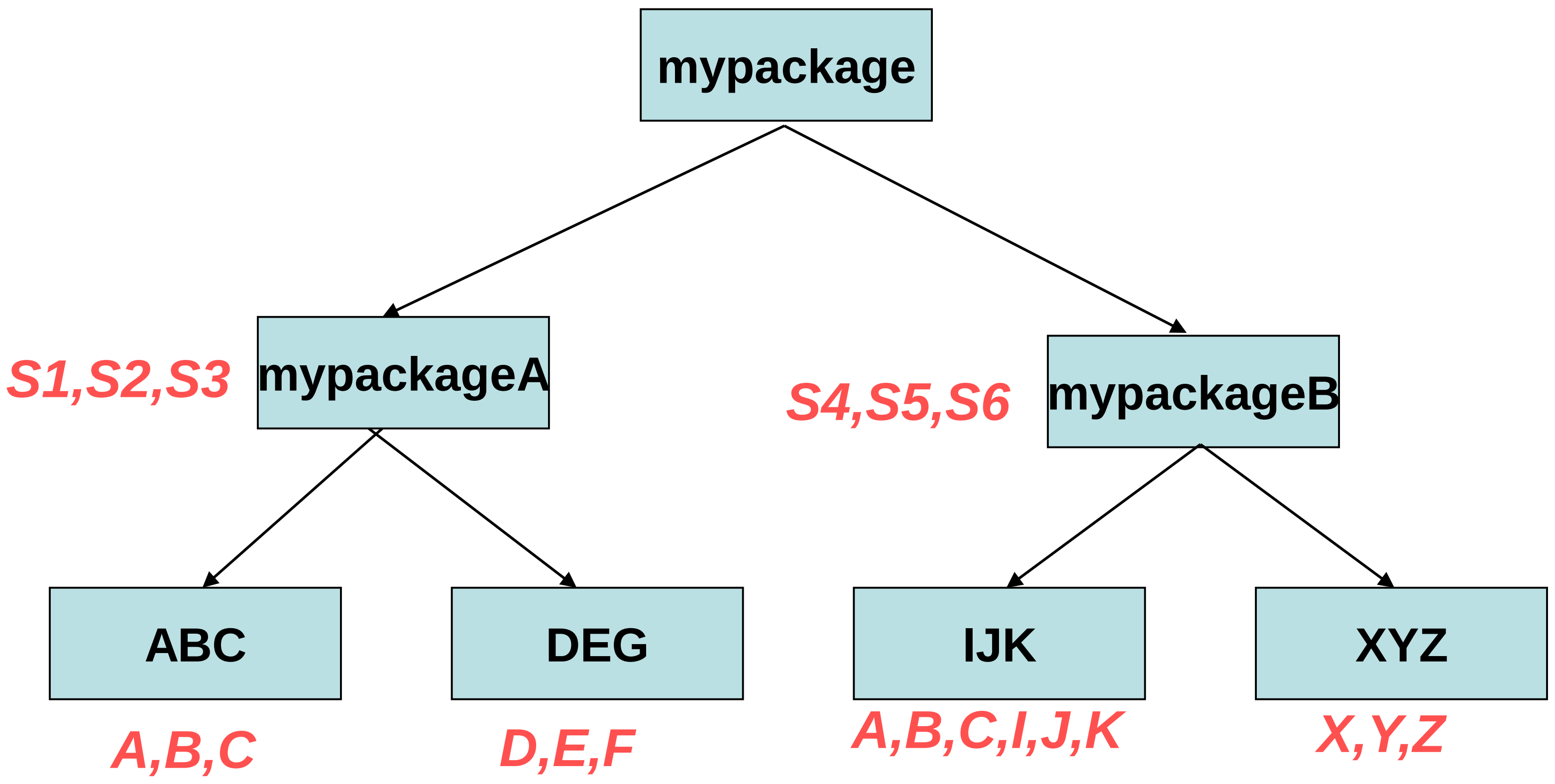
- *Packages enable grouping of functionally related classes*
- *Package names are dot separated, e.g., java.lang.*
- *Package names have a correspondence with the directory structure*
- *Packages Avoid name space collision. There can not be two classes with same name in a same Package. But two packages can have a class with same name.*
- *Exact Name of the class is identifed by its package structure. << Fully Qualified Name>>*

java.lang.String ; *java.util.Arrays;*
java.io.BufferedReader ; *java.util.Date*

Creating Your Own Packages

- *Packages are mirrored through directory structure.*
- *To create a package, First we have to create a directory /directory structure that matches the package hierarchy.*
- *To make a class belongs to a particular package include the <<package>> statement as the first statement of source file.*
- *Note that Outer classes can have public or package-private scope. Nested Classes can have any scope*
- *A source .java file can have only one <<package>> statement. Also note that only one class be public in a source .java file.*

Creating Your Own Package : Example



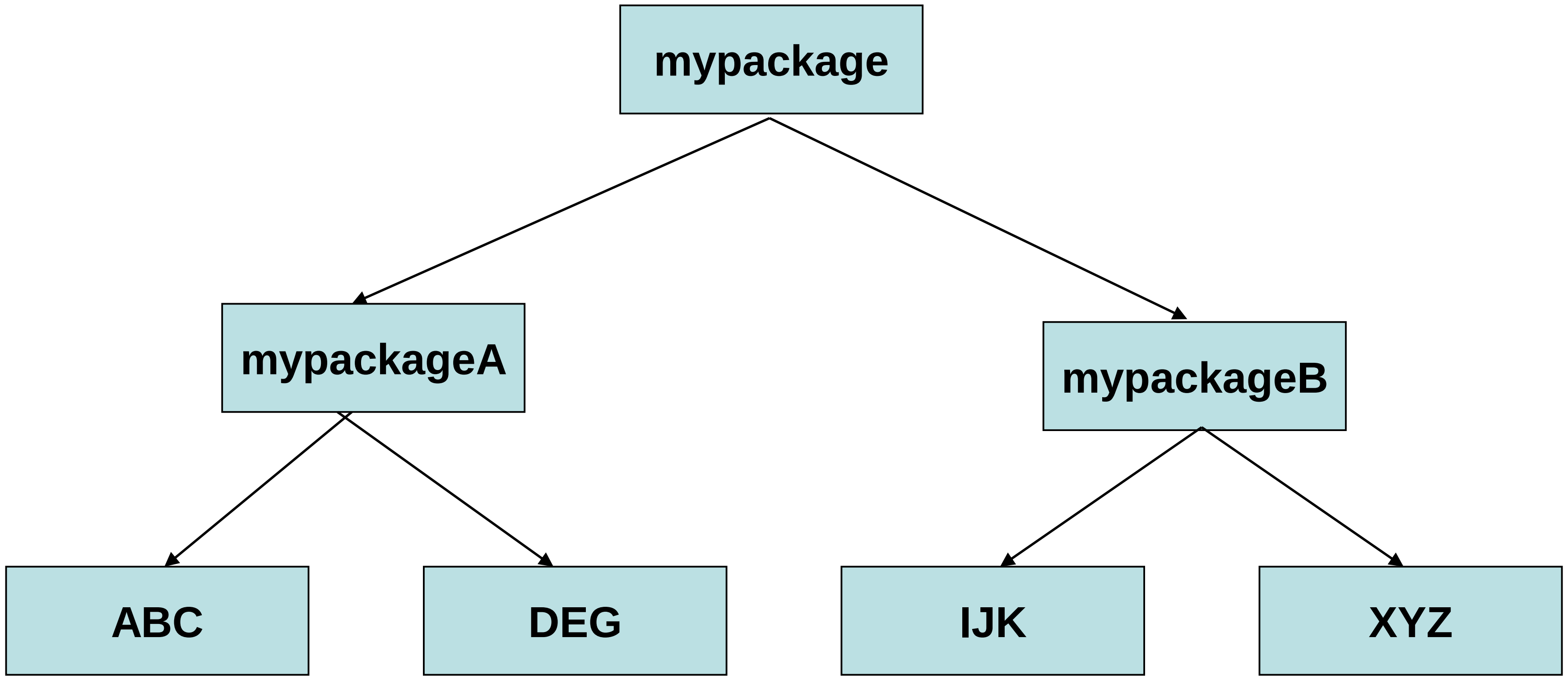
Package ABC and IJK have three classes (A,B and C) with same name.

A class in ABC has name ***mypackage.mypackageA.ABC.A***

A class in IJK has name ***mypackage.mypackageB.IJK.A***

Creating Your Own Package : Example ….

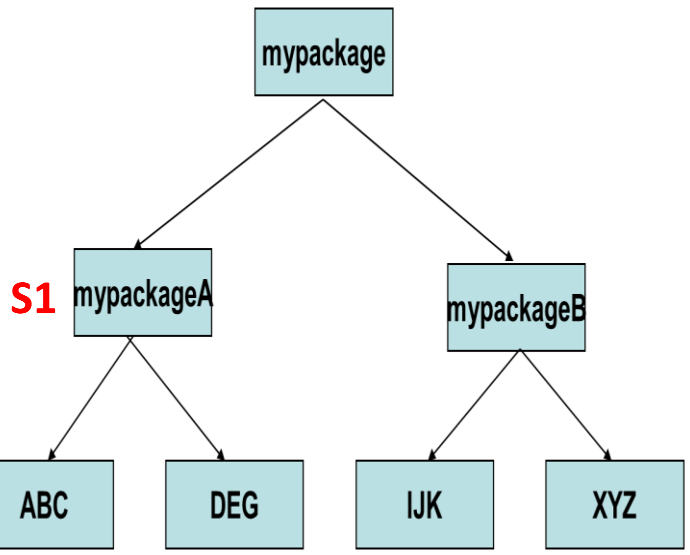
Step 1: Create a Directory Structure That Matches Your Package Structure



Creating Your Own Package : Example ….

Make class S1 belongs to mypackageA with public scope

```
// File Name : S1.java
package mypackage.mypackageA;
public class S1
{
		public S1( )
		{
		System.out.println("This is Class S1");
		} // End of Constructor
} // End of class S1
```

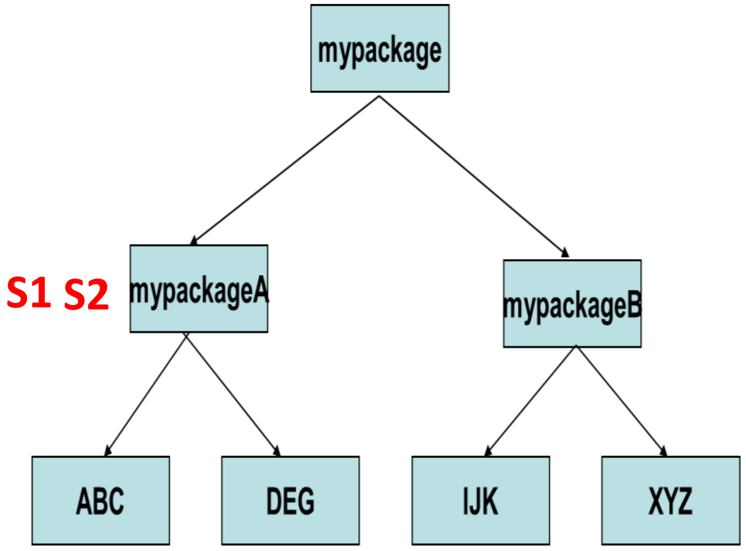


Name the source file as S1.java and compile it.
Store the generated S1.class file in mypackageA directory

Creating Your Own Package : Example ….

Make class S2 belongs to mypackageA with public scope

```
// File Name : S2.java
package mypackage.mypackageA;
public class S2
{
        public S2( )
        {
        System.out.println("This is Class S2");
        } // End of Constructor
} // End of class S2
```

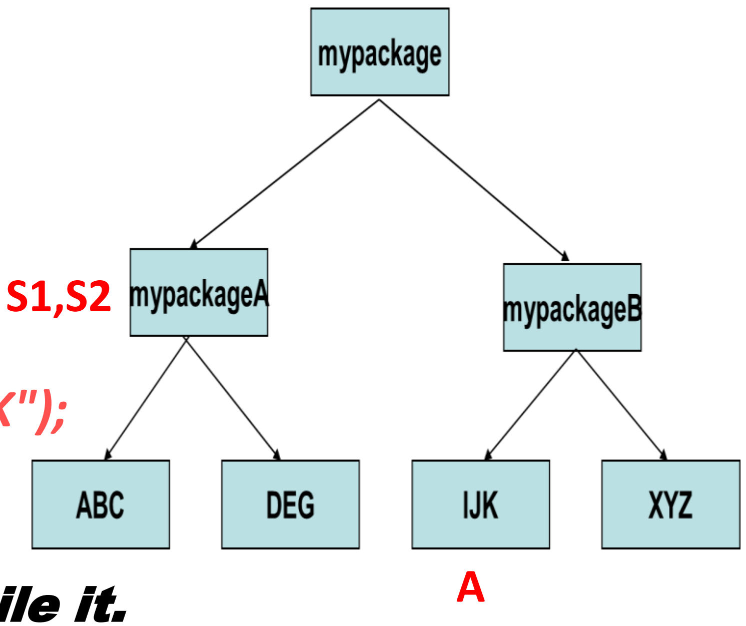


Name the source file as S2.java and compile it.
Store the generated S2.class file in mypackageA directory

Creating Your Own Package : Example ….

Make class A belongs to IJK with public scope

```
// File Name : A.java
package mypackage.mypackageB.IJK;
public class A
{
        public A( )
        {
        System.out.println("This is Class A in IJK");
        } // End of Constructor
} // End of class S2
```



Name the source file as A.java and compile it.
Store the generated A.class file in IJK directory

<< Repeat the Same Procedure for Other classes >>

Importing the Packages

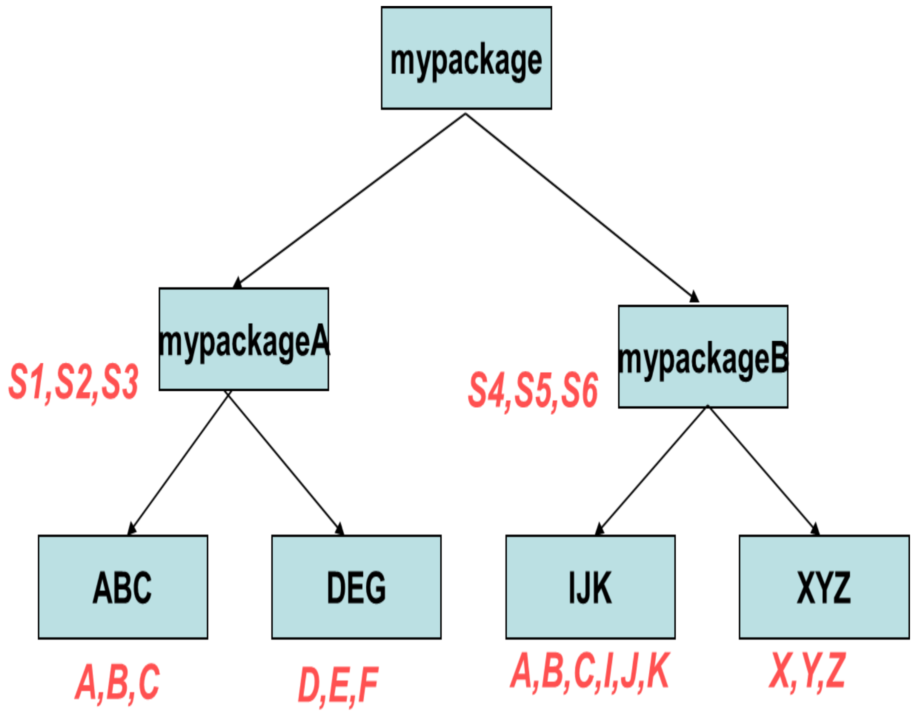
- *import statement allows the importing of package*
- *JRE looks at two places for user created packages*
  1. *Under the current working directory from where you are executing your program*
  2. *At the location specified by CLASSPATH environment variable*
- *Most ideal location for compiling/executing a program is immediately above the package structure.*

Importing the Packages

Suppose you want to use the classes A and B from ABC package

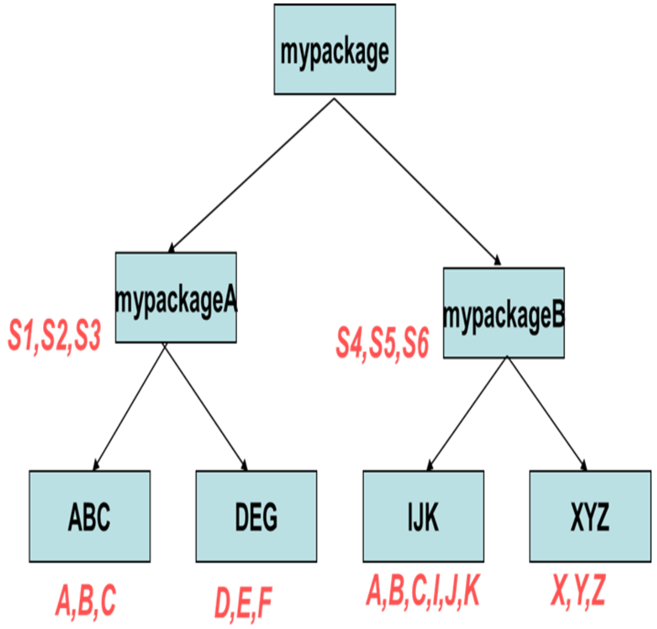
```
// File Name : Test.java
import mypackage.mypackageA.ABC.*;
class Test
{
		public static void main(String args[])
		{
				B b1 = new B();
				C c1 = new C();
		} // End of Method
} // End of class Test
```

Store the Test.java File in a location that is immediately above the package structure i.e. in a folder above mypackage. Compile and Execute it from there.



What is the Error in the Following Code?

```
import mypackage.mypackageA.ABC.*;
Import mypackage.mypackageB.IJK.*;
class Test
{
            public static void main(String args[])
            {
            A a1 = new A();
            } // End of Method
} // End of class Test
```



class A is present in both the imported packages ABC and IJK.
So class A has to referred via its fully qualified name.

mypackage.mypackageA.ABC.A a1 = new mypackage.mypackageA.ABC.A();

OR

mypackage.mypackageB.IJK.A a1 = new mypackage.mypackageB.IJK.A();

Thank You