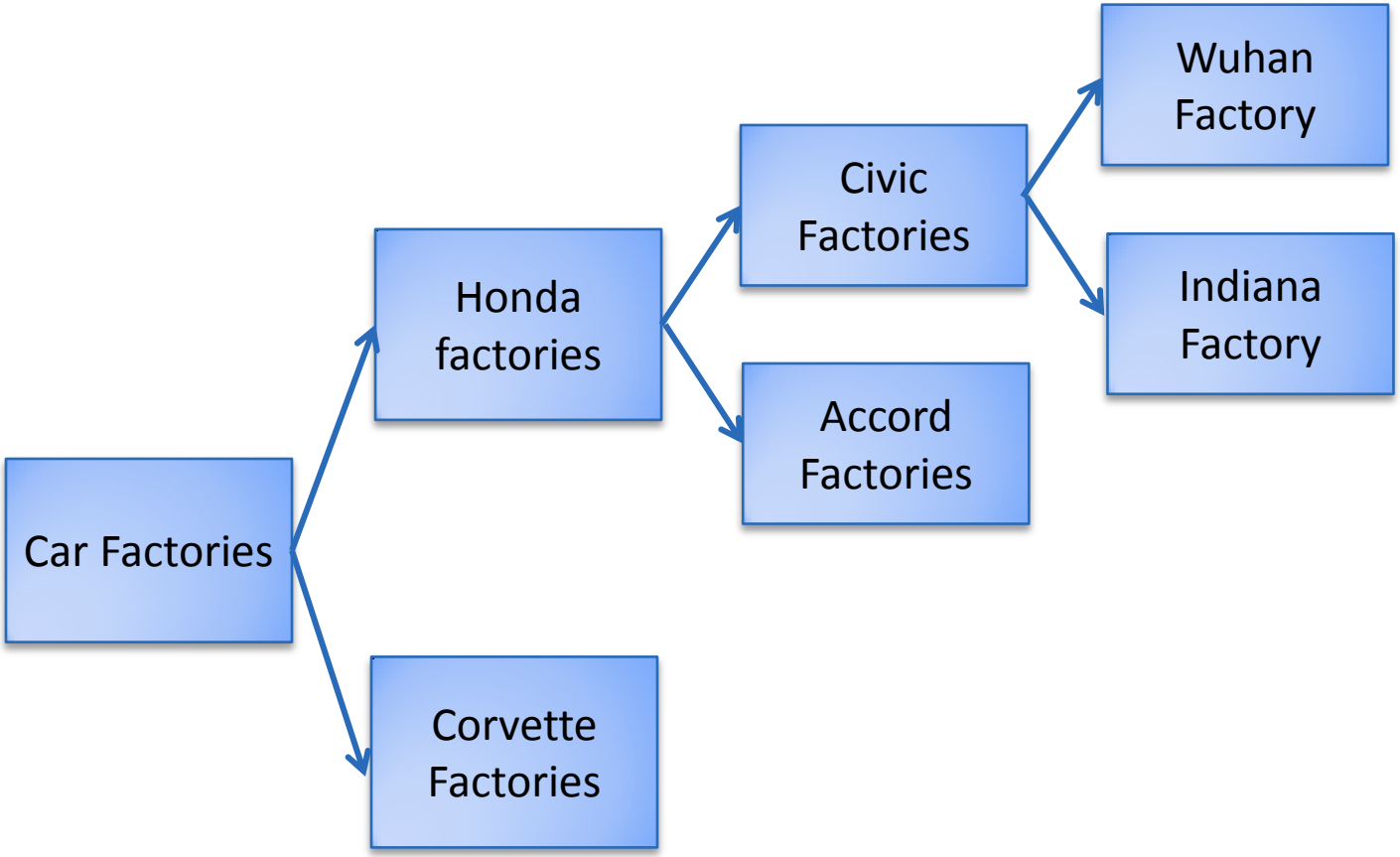
Comp 401
Packages

Instructor: Prasun Dewan

Prerequisites

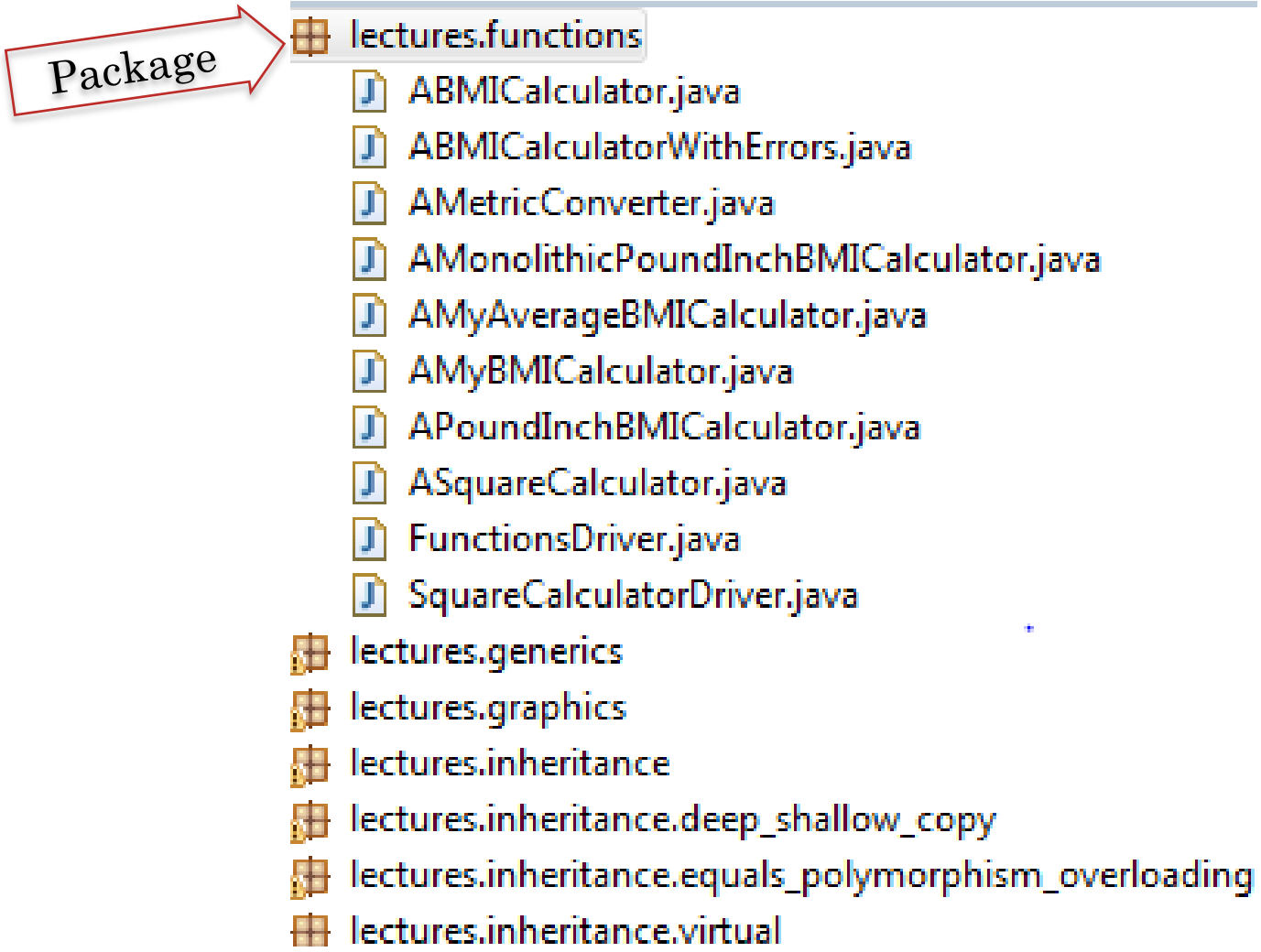
- Objects

Grouping Factories

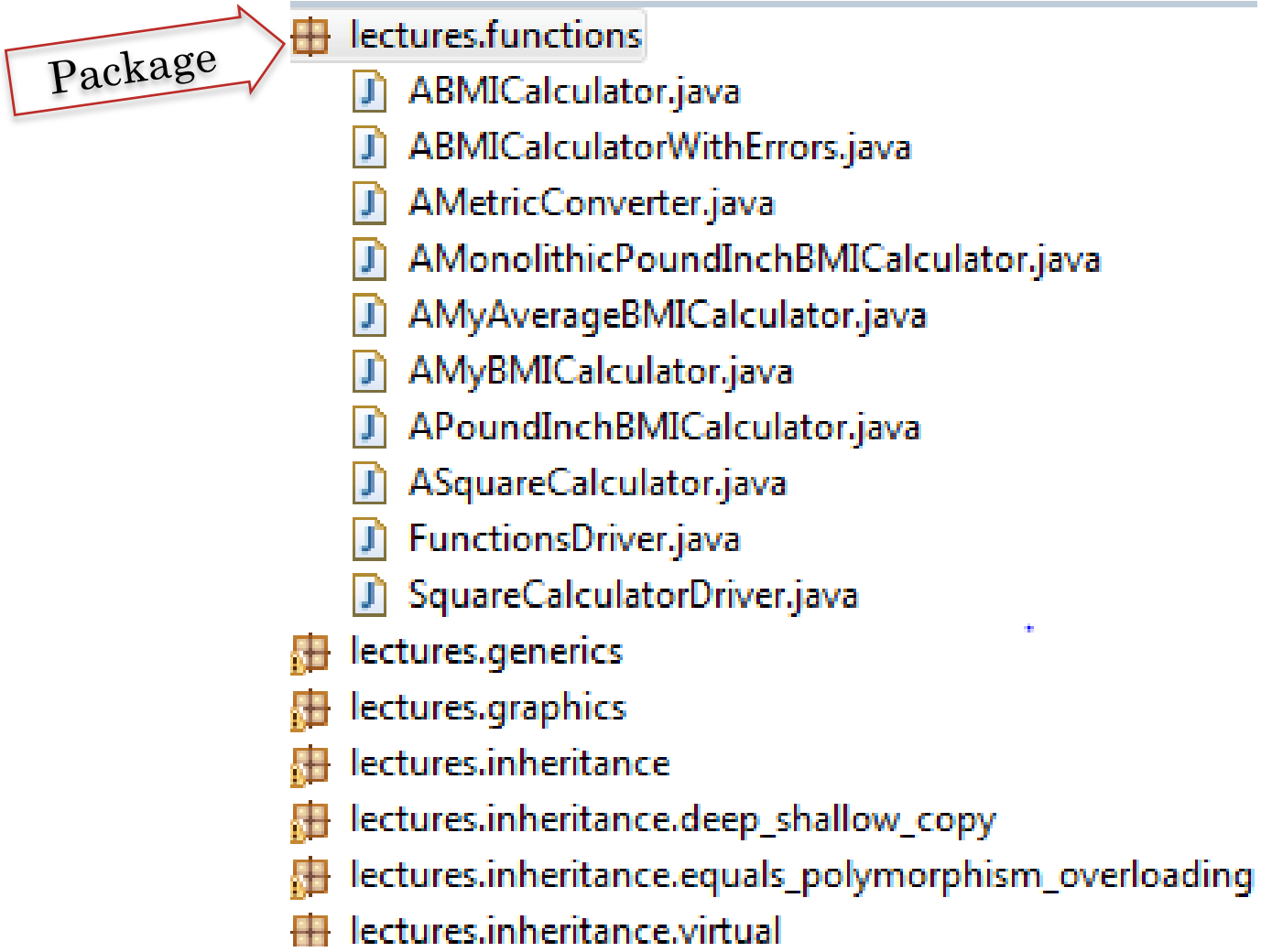


Tablet Factories

Grouping Classes



Grouping Classes



A Simple Instantiated Class

No (default) package

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

```
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Class in different package must be imported using full name of class (a la full file name)

```
package main;
import lectures.functions.ASquareCalculator;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Class in same package need not be imported

```
package lectures.functions;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

No package means package named default, hence import needed

```
import lectures.functions.ASquareCalculator;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

No package means package named default, hence no import needed here

```
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Short name of class in default package same as its full name

```
package main;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Long name with no import

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Can use the full name of class directly

```
package main;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                lectures.functions.ASquareCalculator
squareCalculator = new lectures.functions.ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Long name with no import

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Works but does not tell the reader what is being imported from the package name

```
package main;
import lectures.functions.*;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Important role of import is documentation

Programming style violation

Why imports/full name?

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

```
package lectures.safe_functions;
public class ASquareCalculator
{
        public long square(int x)
        {
                return x*x;
        }
}
```

Twice the size of ints

```
package main;
import lectures.functions.ASquareCalculator;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Disambiguates

Ambiguous Import

```
package lectures.functions;
public class ASquareCalculator
{
	public int square(int x)
	{
		return x*x;
	}
}
```

```
package lectures.safe_functions;
public class ASquareCalculator
{
	public long square(int x)
	{
		return x*x;
	}
}
```

```
package main;
import lectures.functions.ASquareCalculator;
import lectures.safe_functions.ASquareCalculator;
public class SquareCalculatorDriver
{
		public static void main (String[] args) {
			ASquareCalculator squareCalculator = new
ASquareCalculator();
			System.out.println (squareCalculator.square(5));
		}
}
```

Unsuccessful Import

```
package lectures.functions;
class ASquareCalculator
{
        public int square(int x)
                x*x;
```

Non public class usable only within its package

```
package lectures.safe_functions;
public class ASquareCalculator
{
        public long square(int x)
        {
                return x*x;
        }
}
```

```
package main;
import lectures.functions.ASquareCalculator;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

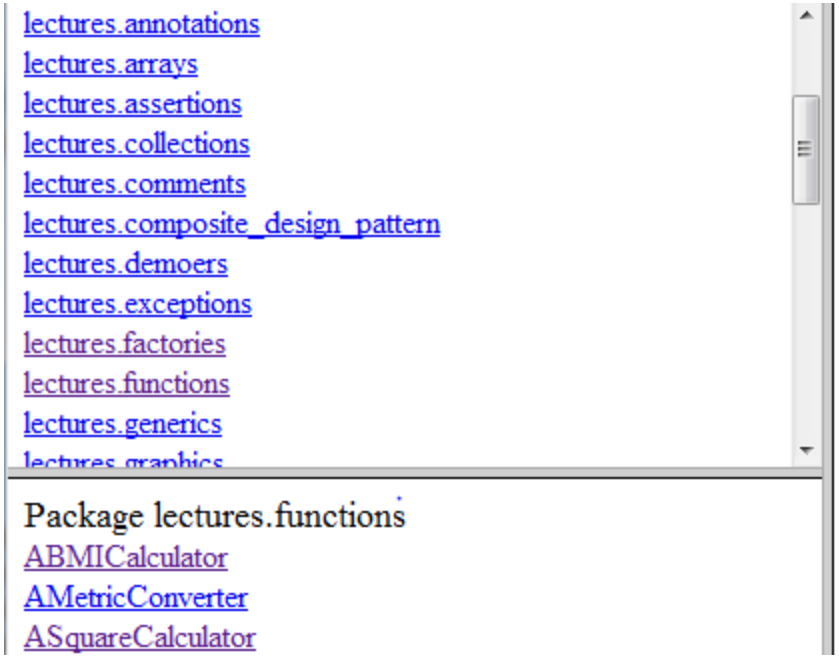
Why Packages?

- Can create competing implementations of same class.
  - A la creating files Test.java in different assignment directories/folders
- Groups related classes together
- Can browse/search for related classes
  - A la browsing through all files in an assignment directory/folder.
- Like directories/folders packages can be hierarchical

  package recitations.functions;

  package lectures.functions;
- Provides documentation of what unrelated classes are being used

Browsing Java Classes

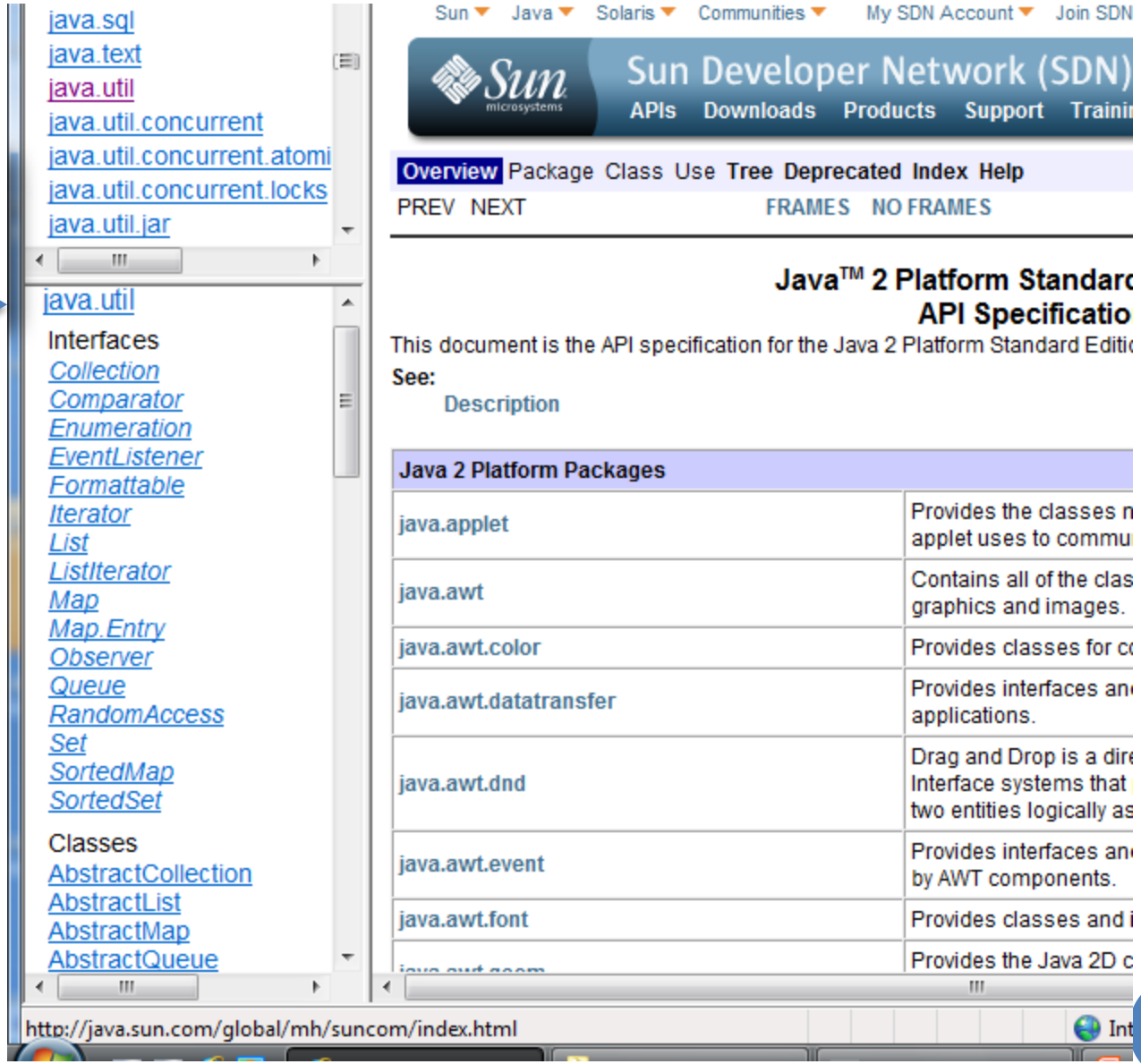



```
package lectures.functions;

public class ASquareCalculator  {
    public int square(int x) {
        return x*x;
    }
}
```

Browsing Java Classes

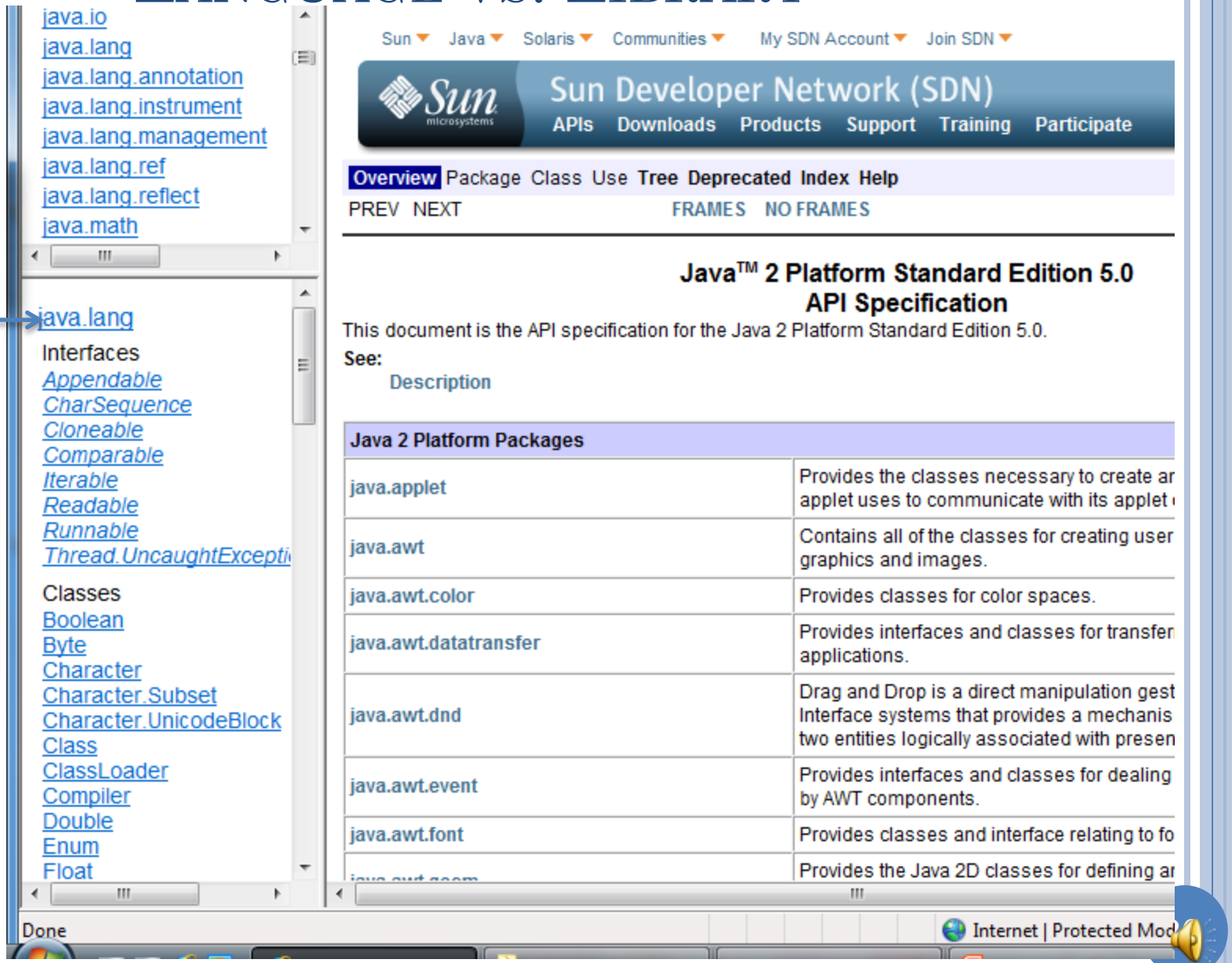
Very useful package



Language vs. Library

Built-in classes

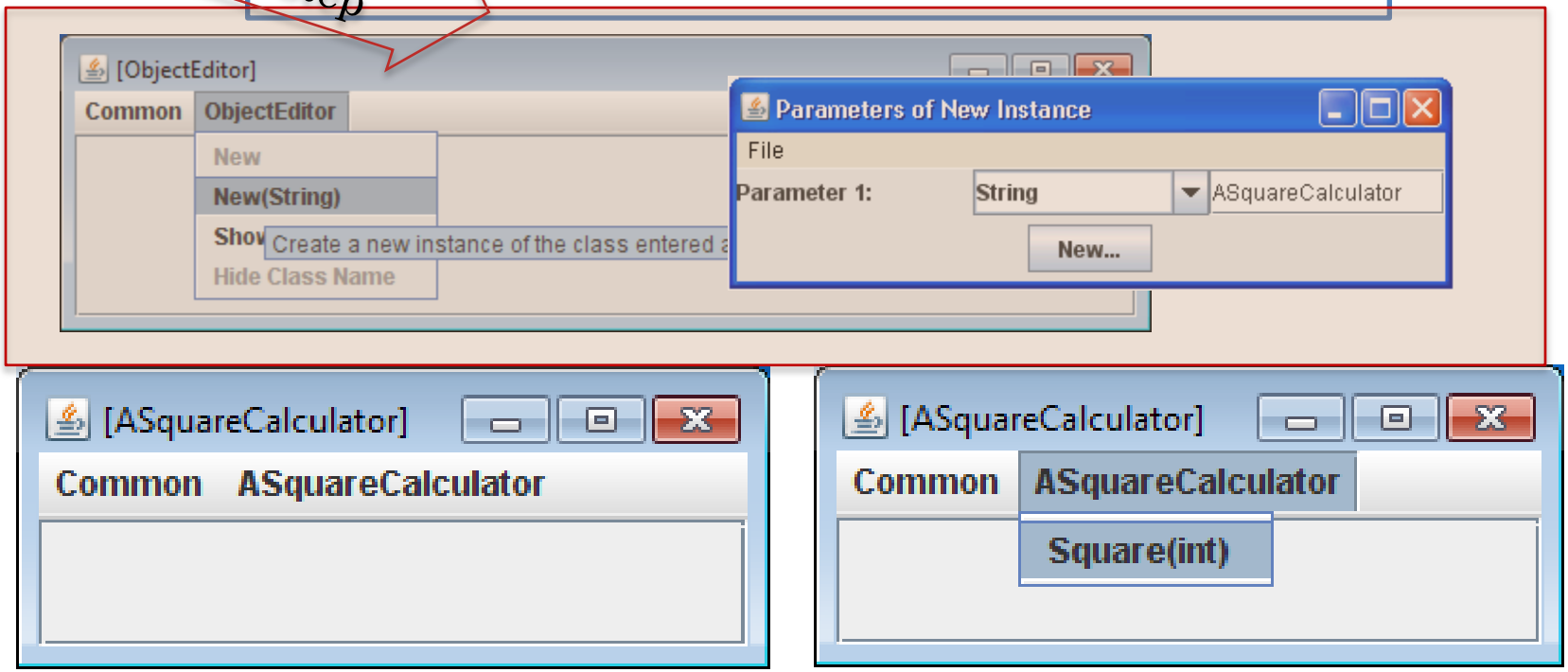
Do not have to be explicitly imported

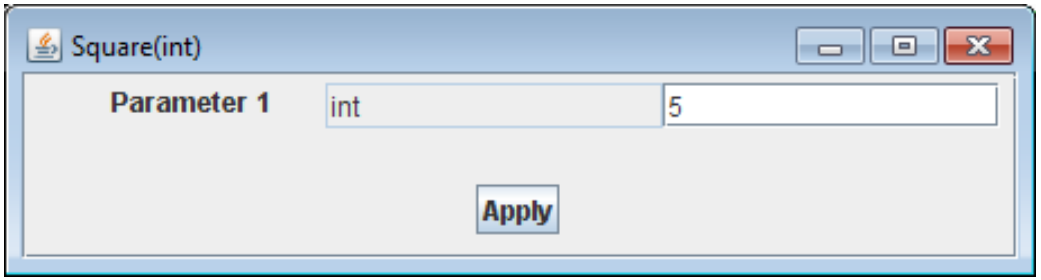


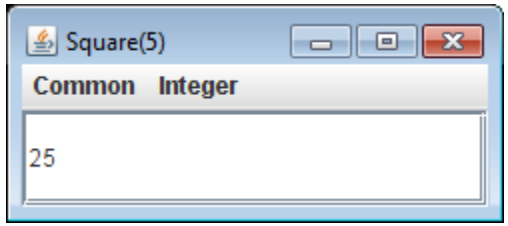
Starting Object Editor Interactively

Can we tell shortcircuit this step

```
java -classpath .;oeall20.jar Comp110ObjectEditor
```







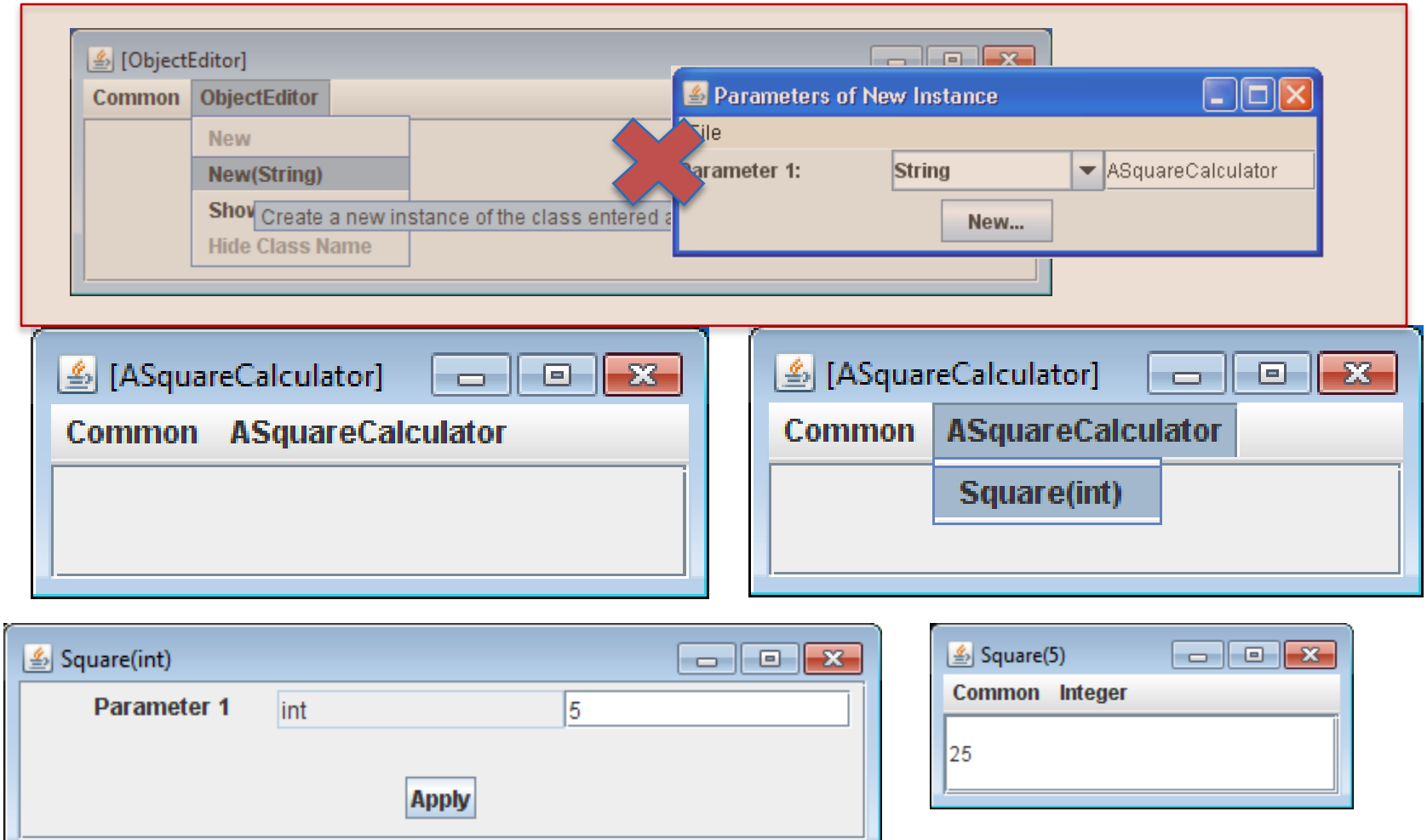
Starting ObjectEditor Programmatically

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

ObjectEditor is predefined packaged class

```
package main;
import lectures.functions.ASquareCalculator;
import bus.uigen.ObjectEditor;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

No Need to Enter Class Name



Remembering Package Names?

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

What if we do not remember the package names?

In Eclipse CTRL_SHIFT_O

```
package main;


public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

Automatic Imports in Eclipse

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

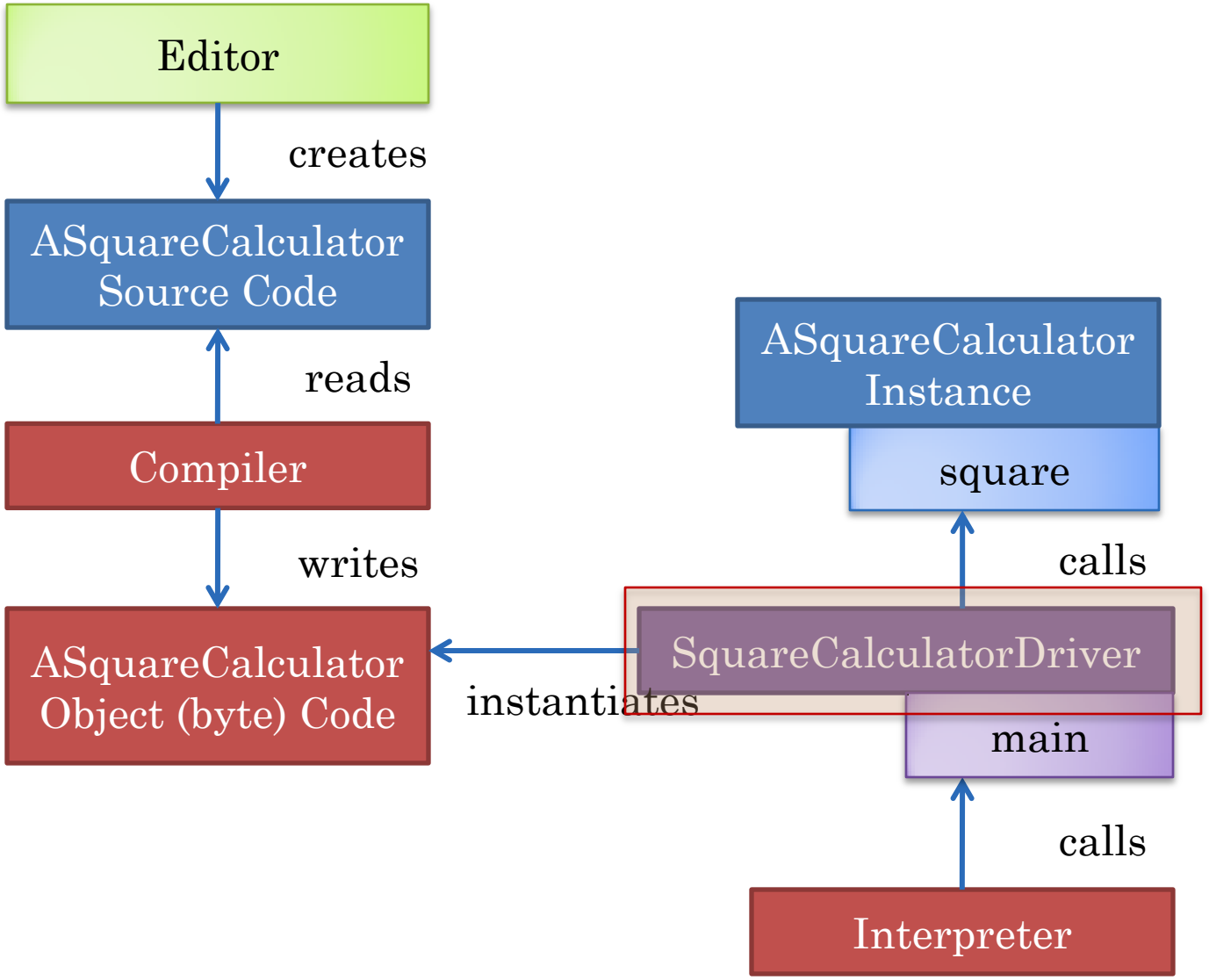
Automatically added

Gives a dialogue box in case multiple classes in same package

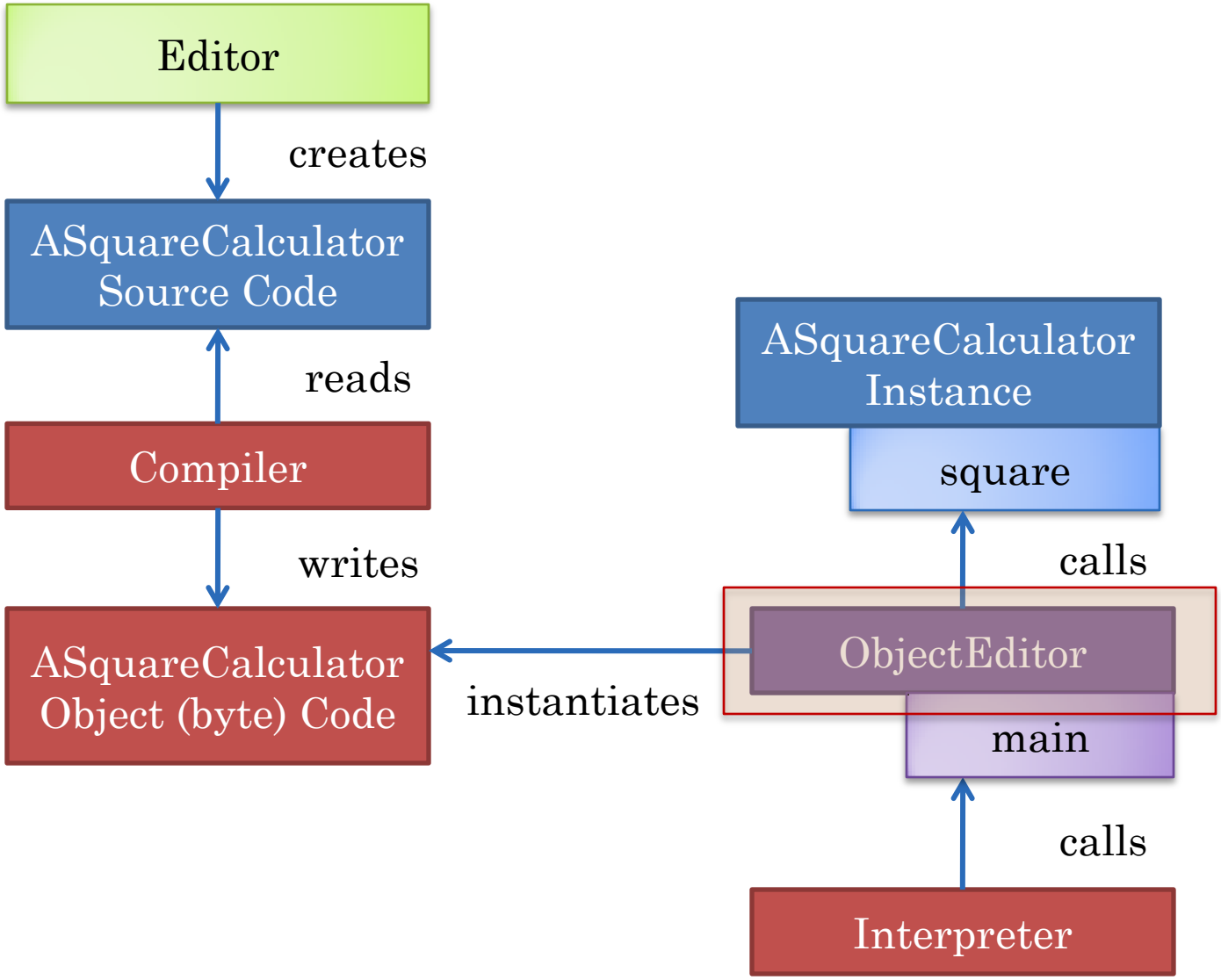
```
package main;
import math.ASquareCalculator;
import bus.uigen.ObjectEditor;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

Package names will not be given in class examples

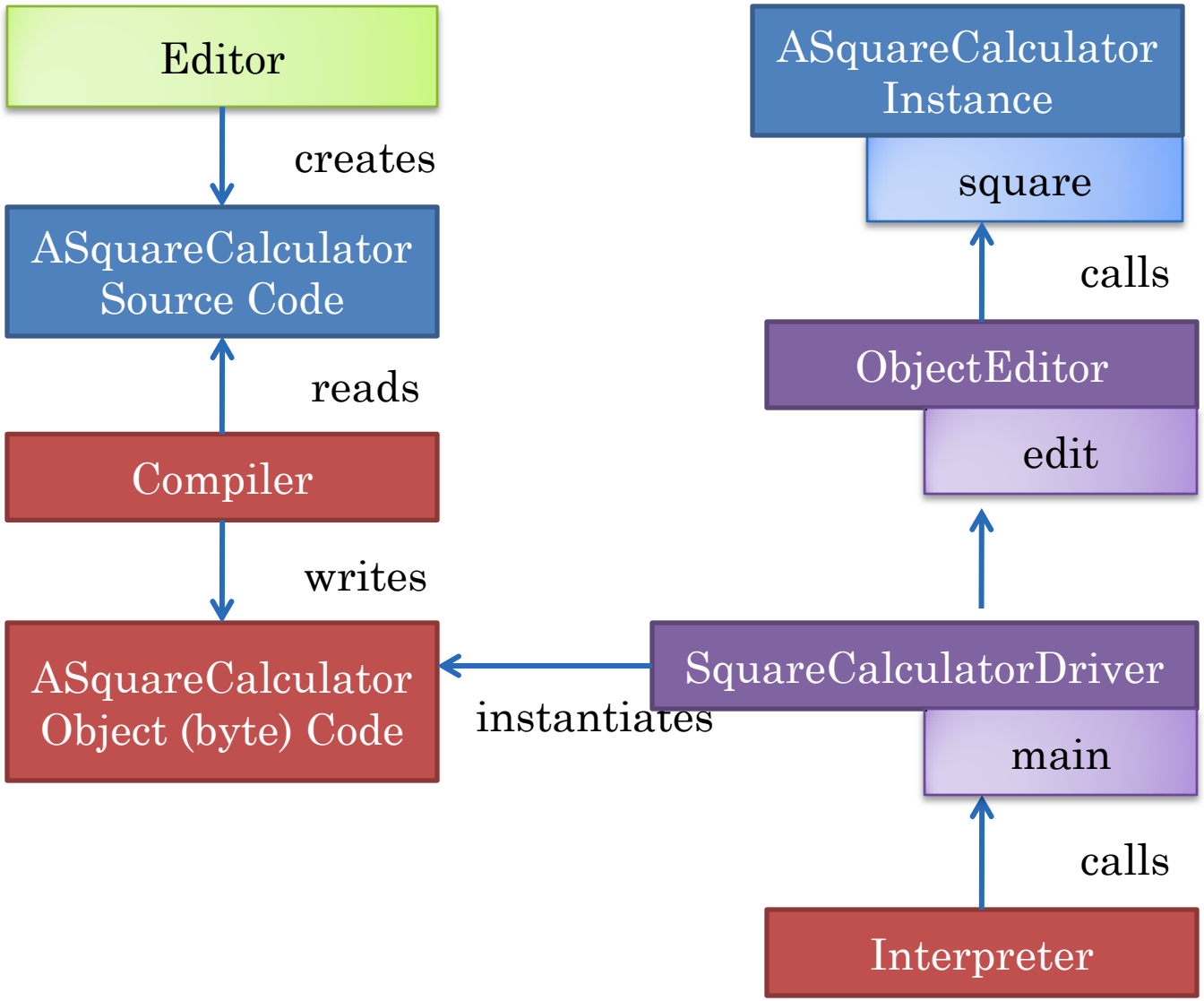
Programmed Main and Method Calls



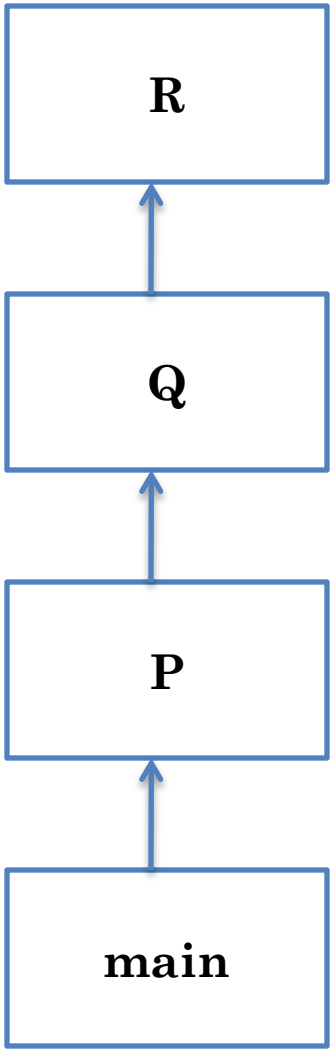
Interactive Main and Method Invocation



Programmed Main and Interactive Method Invocation



Call Chains



Main method starts the computation, and can call other methods.

Can put complete program in main method

Like having one big paragraph in an essay

Method decomposition important modularization technique even in conventional programming

In O-O programming multiple classes involved in computation

Two Classes

```
package lectures.functions;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Can we combine the two classes?

```
package main;
import lectures.functions.ASquareCalculator;
import bus.uigen.ObjectEditor;
public class SquareCalculatorDriver
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

Combining the Classes for Testing

```
package lectures.functions;
import bus.uigen.ObjectEditor;
public class ASquareCalculator  {
  public int square(int x) {
    return x*x;
  }
  public static void main (String[] args) {
    ObjectEditor.edit(new ASquareCalculator());
  }
}
```

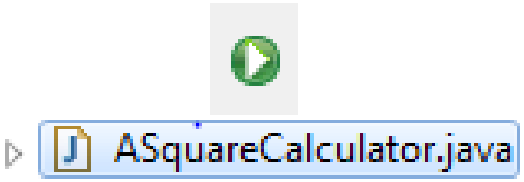
Operation on an instance

Start operations in the factory

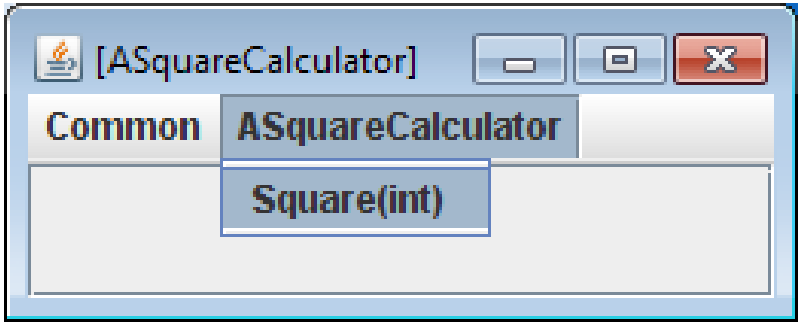
Display the instance

Create a test instance

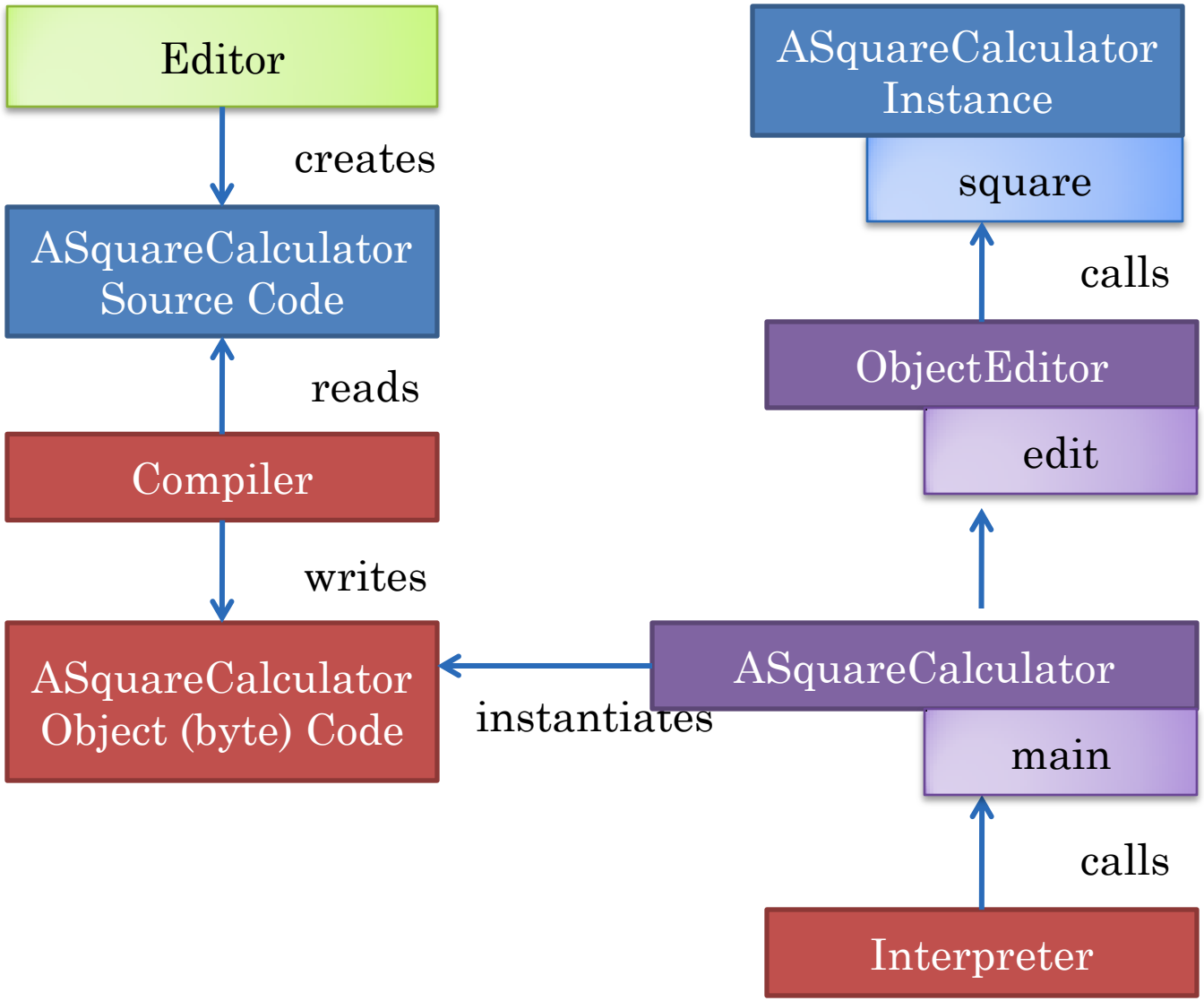
```
java –classpath .;oeall20.jar  ASquareCalculator
```



Convenient but possibly confusing!



Programmed Main and Interactive Method Invocation



Extra Slides

No Package

```
public class ASquareCalculator
{
	public int square(int x)
	{
		return x*x;
	}
}
```

```
public class SquareCalculatorTester
{
		public static void main (String[] args) {
			ASquareCalculator squareCalculator = new
ASquareCalculator();
			System.out.println (squareCalculator.square(5));
		}
}
```

Packages

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Class in different package must be imported using full name of class (a la full file name)

```
package main;
import math.ASquareCalculator;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Class in same package need not be imported

```
package math;

public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

No package means package named default, hence import needed

```
import math.ASquareCalculator;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

No package means package named default, hence no import needed here

```
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Packages

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Short name of class in default package same as its full name

```
package main;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Long name with no import

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Can use the full name of class directly

```
package main;

public class SquareCalculatorTester
{
        public static void main (String[] args) {
                math.ASquareCalculator squareCalculator = new
math.ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Long name with no import

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Works but does not tell the reader what is being imported from the package name

```
package main;
import math.*;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Important role of import is documentation

Programming style violation

Why imports/full name?

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

```
package safemath;
public class ASquareCalculator
{
        public long square(int x)
        {
                return x*x;
        }
}
```

Twice the size of ints

```
package main;
import math.ASquareCalculator;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Disambiguates

Ambiguous Import

```
package math;
public class ASquareCalculator
{
	public int square(int x)
	{
		return x*x;
	}
}
```

```
package safemath;
public class ASquareCalculator
{
	public long square(int x)
	{
		return x*x;
	}
}
```

```
package main;
import math.ASquareCalculator;
import safemath.ASquareCalculator;
public class SquareCalculatorTester
{
	public static void main (String[] args) {
		ASquareCalculator squareCalculator = new
ASquareCalculator();
		System.out.println (squareCalculator.square(5));
	}
}
```

Unsuccessful Import

```
package math;
class ASquareCalculator
{
        public int square(int x)
                x*x;
```

Non public class usable only within its package

```
package safemath;
public class ASquareCalculator
{
        public long square(int x)
        {
                return x*x;
        }
}
```

```
package main;
import math.ASquareCalculator;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Why Packages?

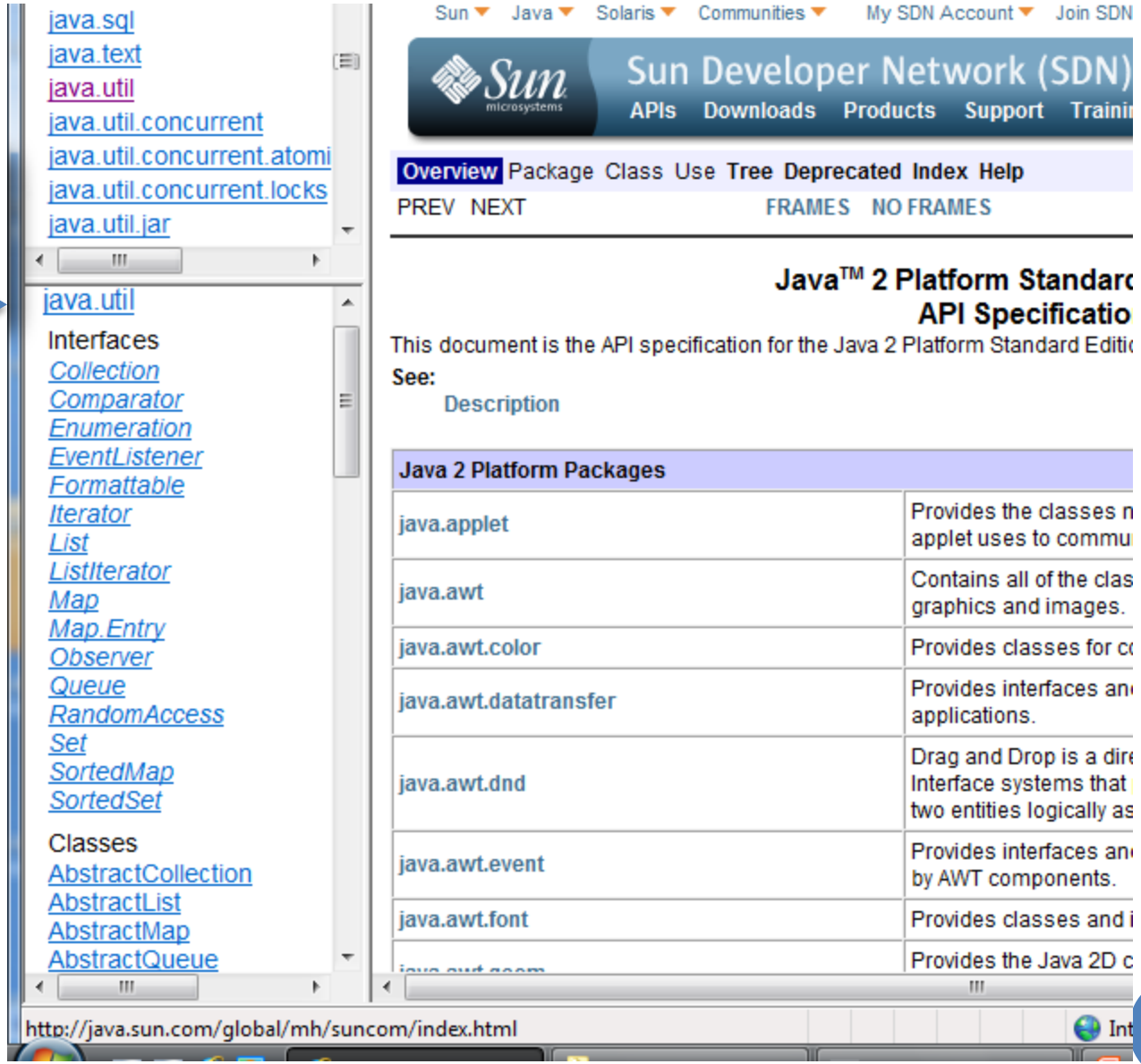
- Can create competing implementations of same class.
  - A la creating files Test.java in different assignment directories/folders
- Groups related classes together
- Can browse/search for related classes
  - A la browsing through all files in an assignment directory/folder.
- Like directories/folders packages can be hierarchical

  package math.power;

  public class ACubeCalculator {…}
- Provides documentation of what unrelated classes are being used

Browsing Java Classes

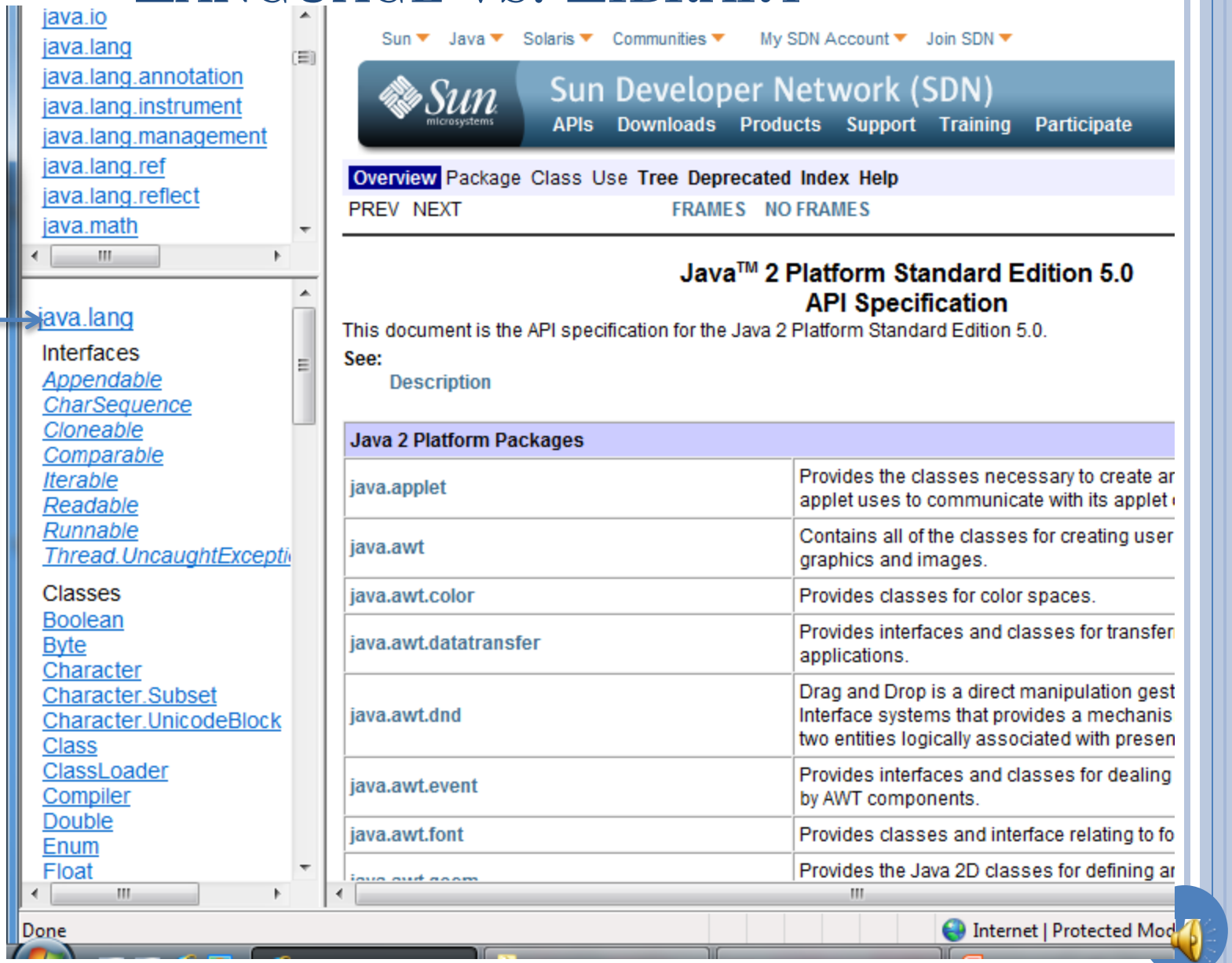
Very useful package



Language vs. Library

Built-in classes

Do not have to be explicitly imported



Changing Parameter

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Calculates 5*5

```
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(5));
        }
}
```

Changing Parameter

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Must change code

```
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println (squareCalculator.square(341));
        }
}
```

Rerun Program

How to not re-run program without writing tedious UI code?

```
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Must re-run program

```
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                System.out.println
(squareCalculator.square(Integer.parseInt(args[0]));
        }
}
```

ObjectEditor

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

ObjectEditor is predefined packaged class

```
package main;
import math.ASquareCalculator;
import bus.uigen.ObjectEditor;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

Remembering Package Names?

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

What if we do not remember the package names?

In Eclipse CTRL_SHIFT_O

```
package main;


public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

Automatic Imports in Eclipse

```
package math;
public class ASquareCalculator
{
        public int square(int x)
        {
                return x*x;
        }
}
```

Automatically added

Gives a dialogue box in case multiple classes in same package

```
package main;
import math.ASquareCalculator;
import bus.uigen.ObjectEditor;
public class SquareCalculatorTester
{
        public static void main (String[] args) {
                ASquareCalculator squareCalculator = new
ASquareCalculator();
                ObjectEditor.edit(squareCalculator );
        }
}
```

Package names will not be given in class examples

Summary: Java vs. Real-world

Java	Real-world
Class	Factory
Computer Object	Manufactured Physical Object
Method	Operation
Invoking/Executing a Method	Performing an Operation
Instance of a Class	Manufactured by a Factory
Defining/Declaring a Class	Constructing a Factory
Instantiating a Class	Manufacturing an Object
Class (Static) Method	Operation on a Factory
Main Class Method	Initiates computation
Instance (Non static) method	Operation on an instance of a class
Instance (Non static) method	Grouping of factories by states, country