Delivering Software Components (SC)

Never Stand Still | Faculty of Engineering | CSE

System Development

- Every system S consists of one or more software components $\{c1, \ldots .\, cn\}$
- A Software Component C encompasses :
  - Set of related functions $\{f1, \ldots .\, fn\}$
  - Well defined interface I

  ➔Each Interface I consists of a set of parameters $\{p1 \ldots pn\}$

Components Development

- To develop/support a component C we need to satisfy it's related functions $\{f1, \dots . fn\}$and interface I, to do so we need to:

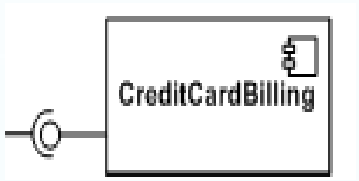
➔Understand the business requirements document (Assignment Spec)

➔ Deliver Technical specifications document (how the business requirements will be met)

Characteristics of SC

Encapsulation

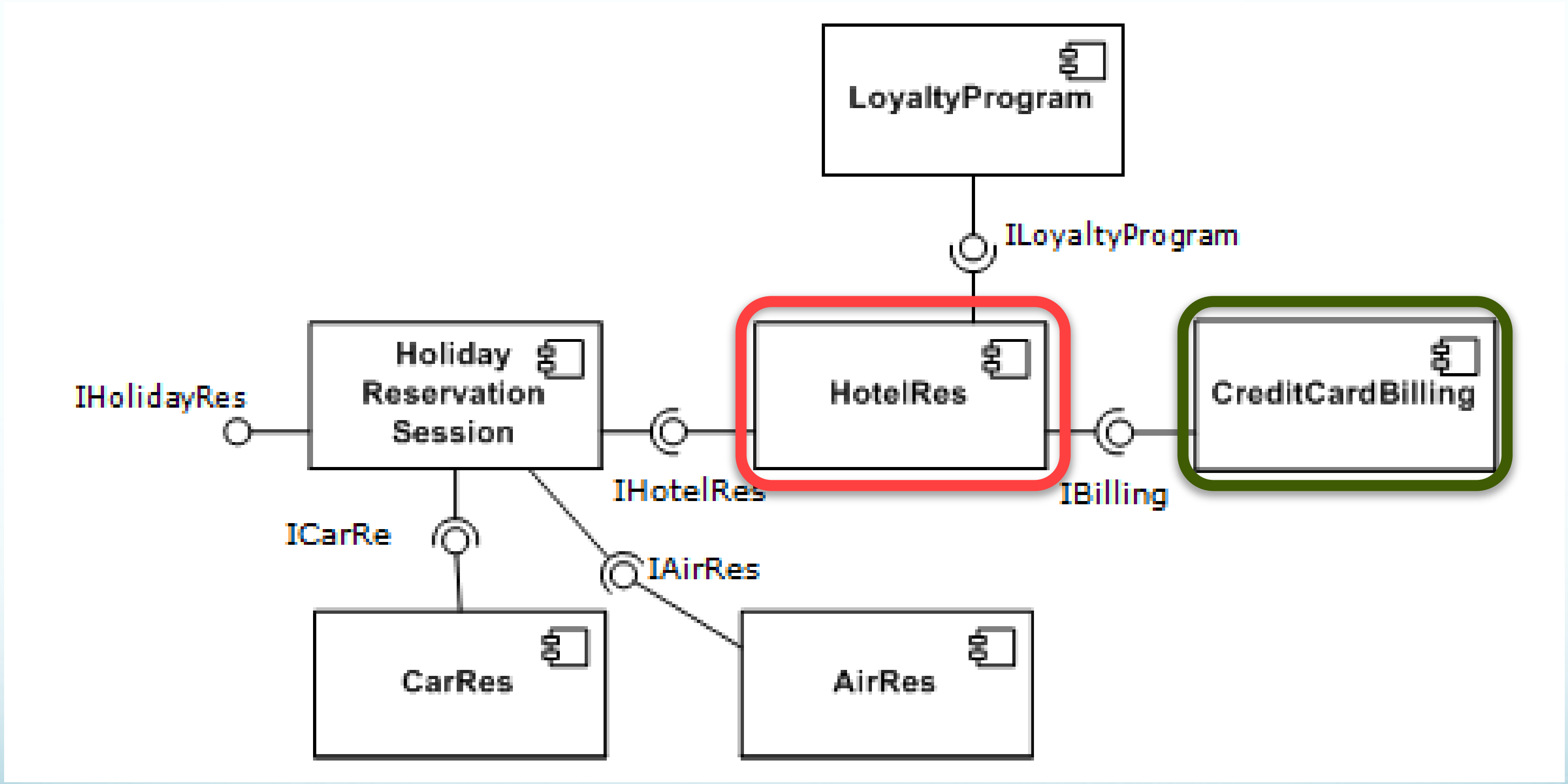
Interface



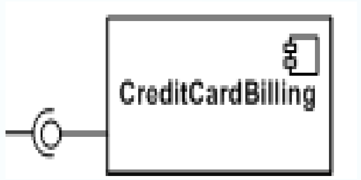
Replaceable

Reusable

Software Component Example



Why SC are good?



- **For component provider**
  - Able to change the implementation of the component as long as the **interface** is still satisfied
  - New requirements can be delivered as new components, without having to change the existing components
- **For application builder**
  - Don’t need to recompile/redeploy anything(with the same **interface** and functionality)
  - No need to understand the inner working, but only the **interface** of the component

Components are like black boxes

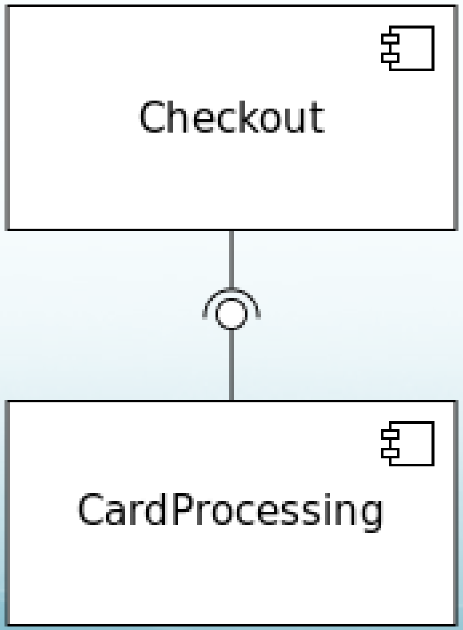
- The programmer ***knows***:
  - how the outside looks like
  - what the component can provide
- The programmer ***does NOT know***:
  - how it works internally

Developing Software Components

- DOS Batch file
- Linux/Unix Shell Script
- C# DLL file
- JAR File
- Executable file (.exe)
- Web service (e.g. REST)

Examples

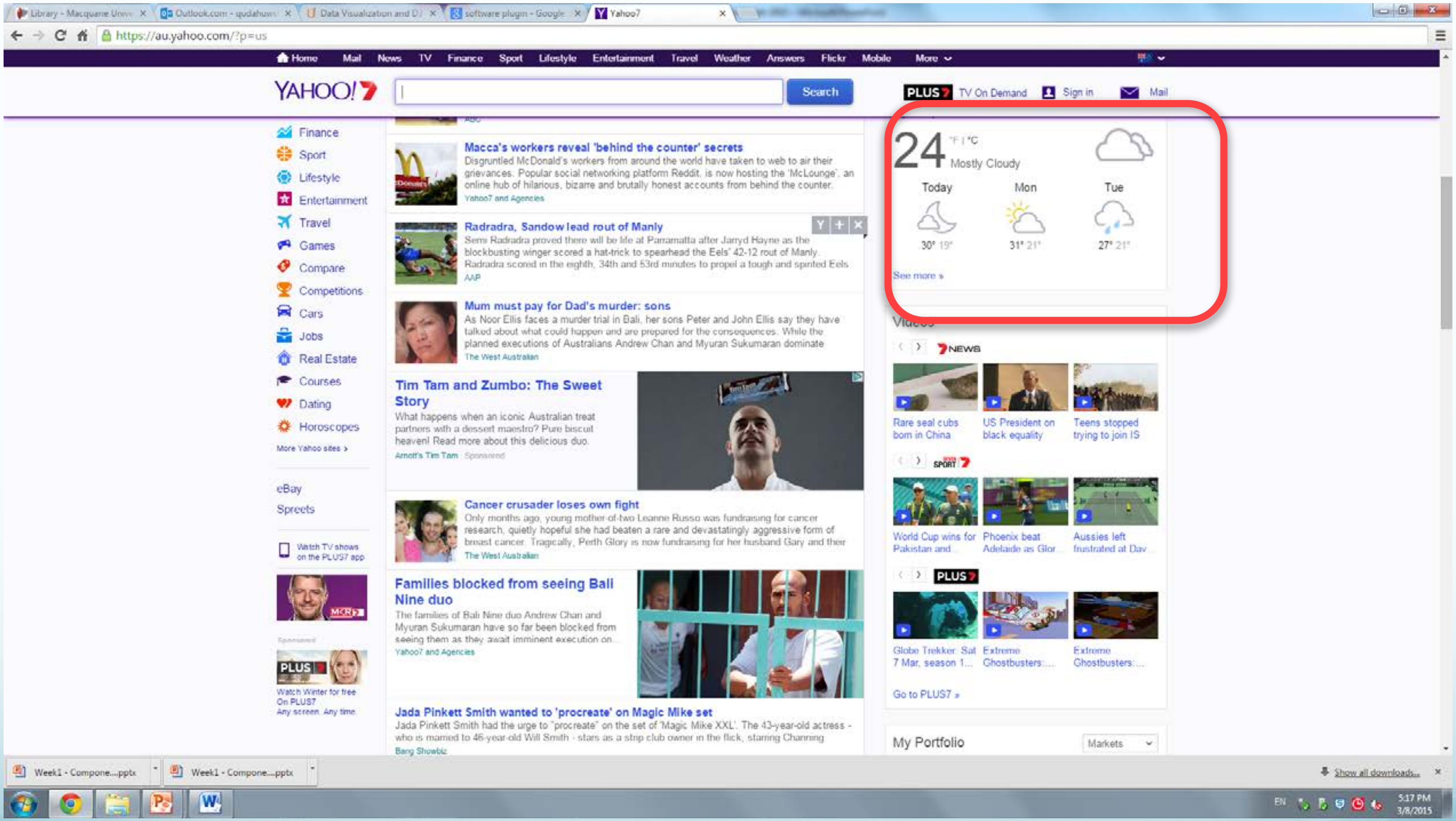
- A small interest calculator plug-in
- An interface to a database manager
- Paypal:



Ebay checkout app

Paypal service

Weather Plug-in



24 °F | °C Mostly Cloudy

Today	Mon	Tue
30° 19°	31° 21°	27° 21°

See more »

Macca's workers reveal 'behind the counter' secrets
Disgruntled McDonald's workers from around the world have taken to web to air their grievances. Popular social networking platform Reddit, is now hosting the 'McLounge', an online hub of hilarious, bizarre and brutally honest accounts from behind the counter.
Yahoo7 and Agencies

Radradra, Sandow lead rout of Manly
Semi Radradra proved there will be life at Parramatta after Jarryd Hayne as the blockbusting winger scored a hat-trick to spearhead the Eels' 42-12 rout of Manly. Radradra scored in the eighth, 34th and 53rd minutes to propel a tough and spirited Eels
AAP

Mum must pay for Dad's murder: sons
As Noor Ellis faces a murder trial in Bali, her sons Peter and John Ellis say they have talked about what could happen and are prepared for the consequences. While the planned executions of Australians Andrew Chan and Myuran Sukumaran dominate
The West Australian

Tim Tam and Zumbo: The Sweet Story
What happens when an iconic Australian treat partners with a dessert maestro? Pure biscuit heaven! Read more about this delicious duo.
Arnott's Tim Tam Sponsored

Cancer crusader loses own fight
Only months ago, young mother-of-two Leanne Russo was fundraising for cancer research, quietly hopeful she had beaten a rare and devastatingly aggressive form of breast cancer. Tragically, Perth Glory is now fundraising for her husband Gary and their
The West Australian

Families blocked from seeing Bali Nine duo
The families of Bali Nine duo Andrew Chan and Myuran Sukumaran have so far been blocked from seeing them as they await imminent execution on...
Yahoo7 and Agencies

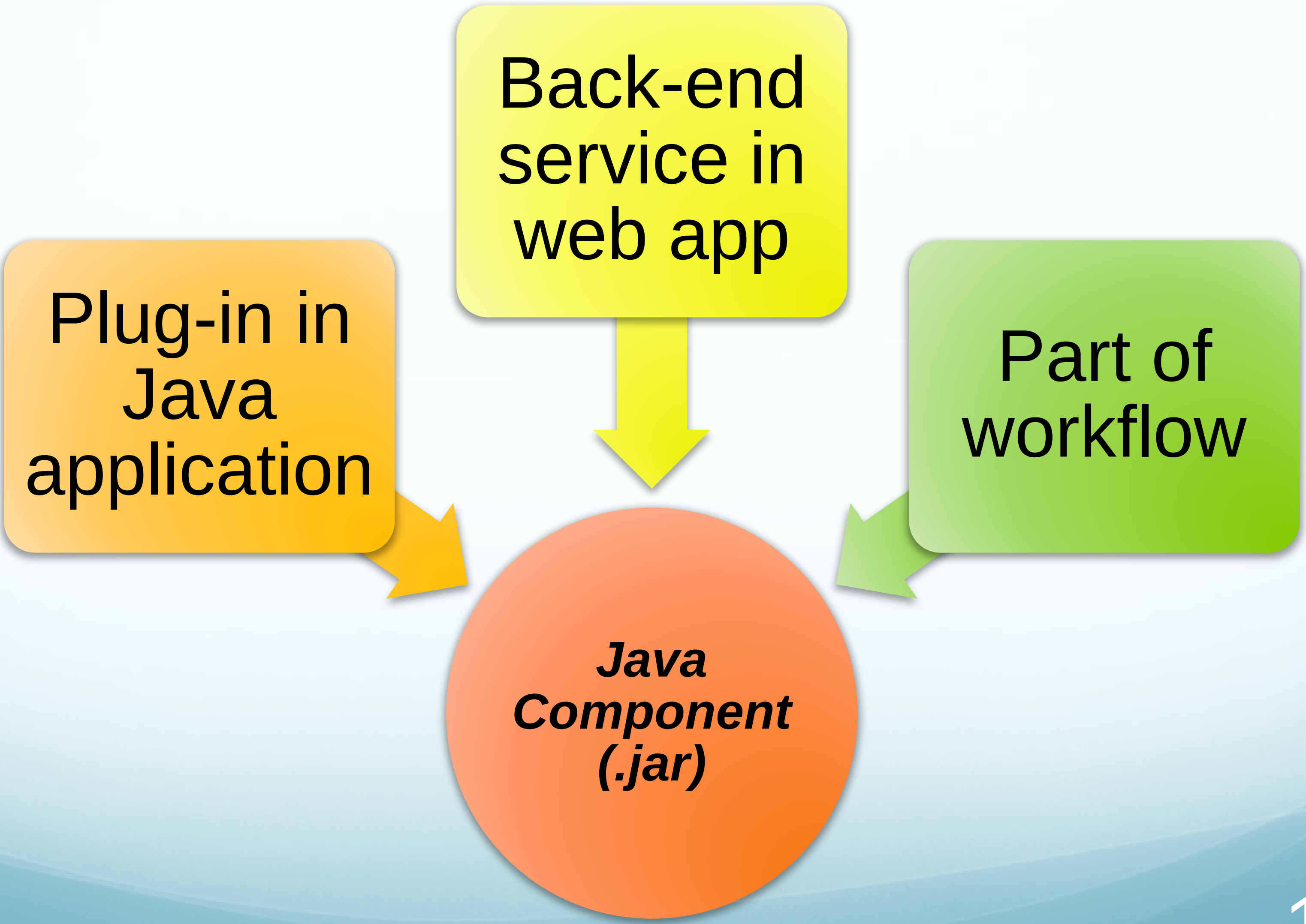
Jada Pinkett Smith wanted to 'procreate' on Magic Mike set
Jada Pinkett Smith had the urge to "procreate" on the set of 'Magic Mike XXL'. The 43-year-old actress - who is married to 46-year-old Will Smith - stars as a strip club owner in the flick, starring Channing
Bang Showbiz

Rare seal cubs born in China
US President on black equality
Teens stopped trying to join IS
World Cup wins for Pakistan and...
Phoenix beat Adelaide as Glor...
Aussies left frustrated at Dav...
Globe Trekker: Sat 7 Mar, season 1...
Extreme Ghostbusters:...
Extreme Ghostbusters:...
Go to PLUS7 »

Calling a component

- Java component
- C# component
- REST component
- From a workflow – like Taverna Workflow

Java component

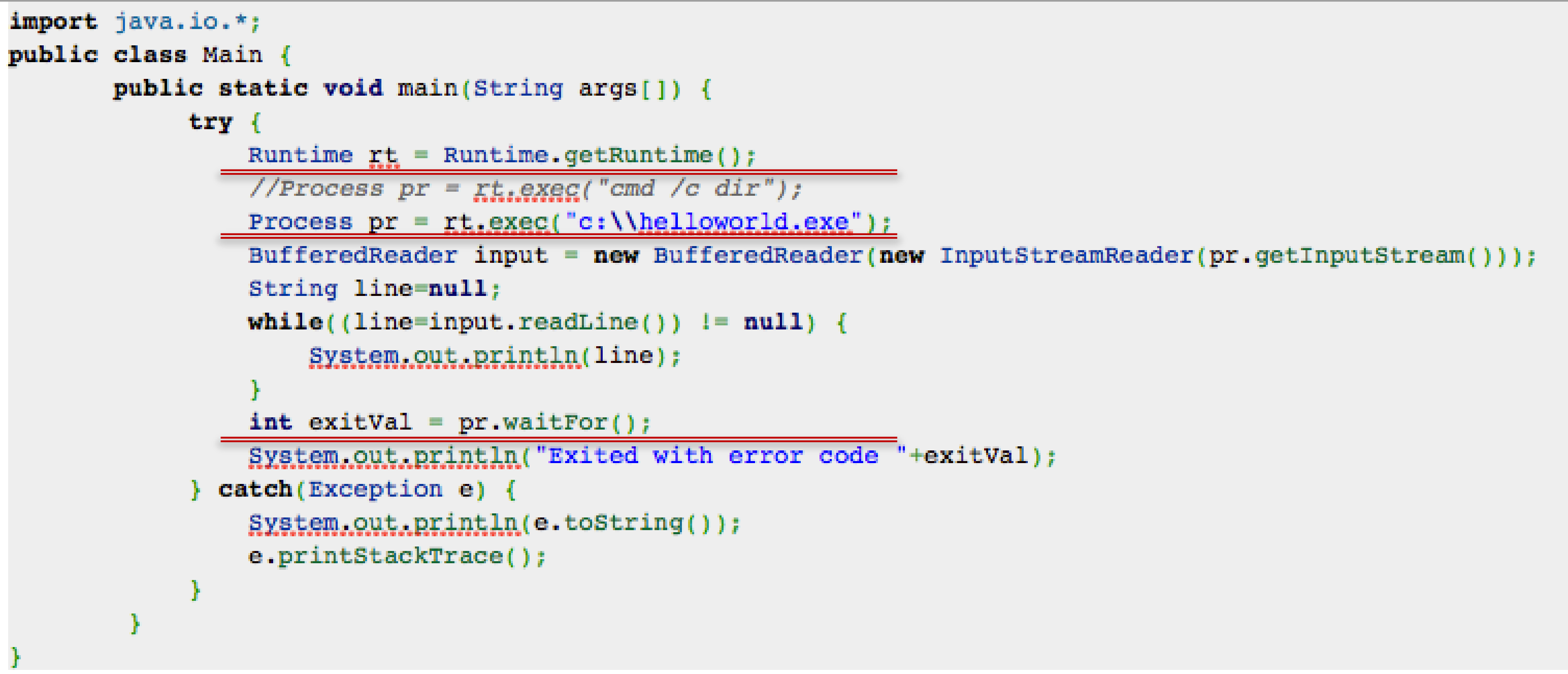


How to Generate **.jar** File

- Export from IDE (e.g. Eclipse)
- Use command line:
  - jar cf jar-file input-file(s)
- Use popular build tools:
  - Maven
  - Ant
  - Buildr
  - … …

Call Dos/Linux commands from Java

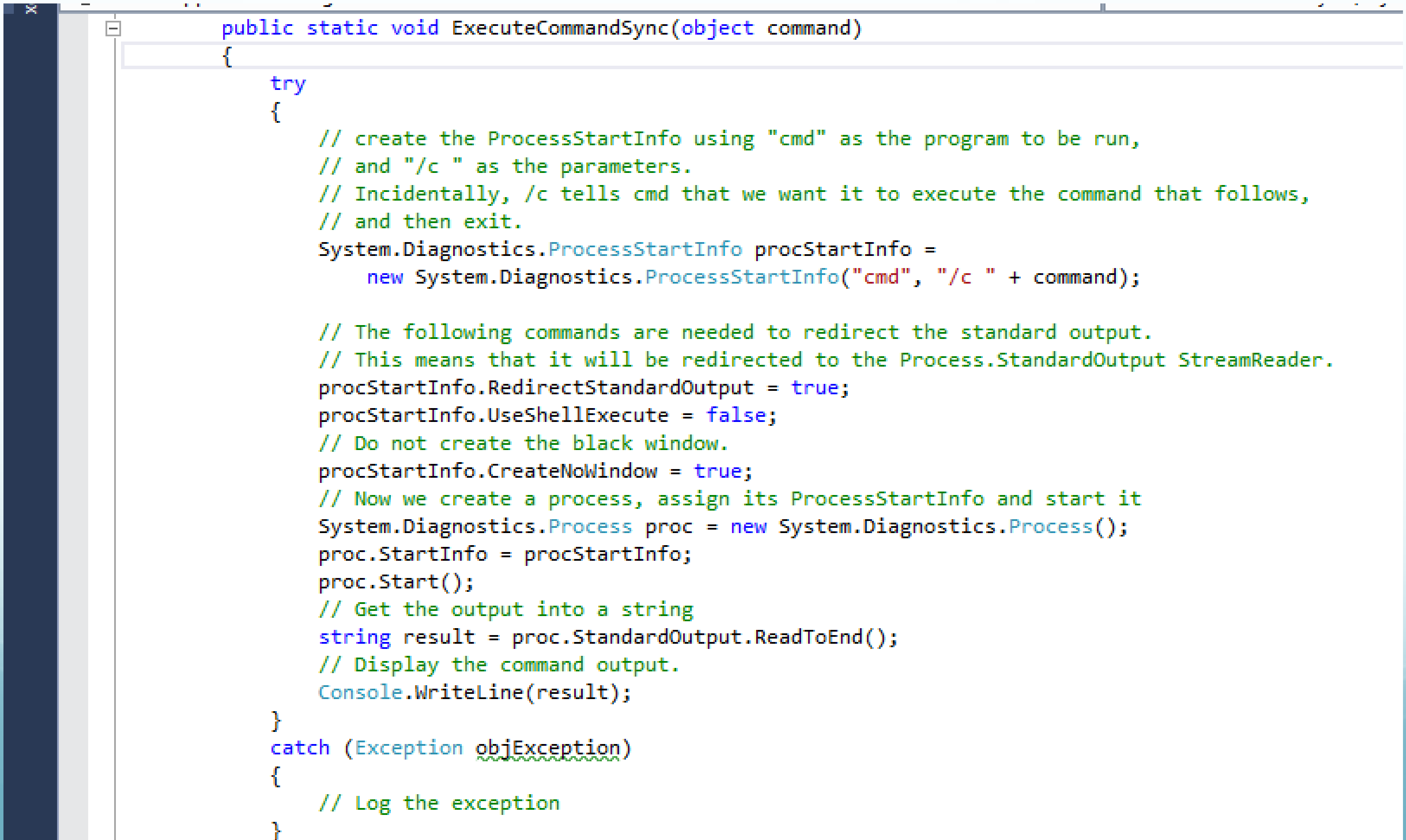
- ***Process*** or ***ProcessBuilder***



```
import java.io.*;
public class Main {
       public static void main(String args[]) {
            try {
                Runtime rt = Runtime.getRuntime();
                //Process pr = rt.exec("cmd /c dir");
                Process pr = rt.exec("c:\\helloworld.exe");
                BufferedReader input = new BufferedReader(new InputStreamReader(pr.getInputStream()));
                String line=null;
                while((line=input.readLine()) != null) {
                    System.out.println(line);
                }
                int exitVal = pr.waitFor();
                System.out.println("Exited with error code "+exitVal);
            } catch(Exception e) {
                System.out.println(e.toString());
                e.printStackTrace();
            }
        }
}
```

Execute Batch file From C# [4]

- Invoke DOS batch file from C#



```
        public static void ExecuteCommandSync(object command)
        {
            try
            {
                // create the ProcessStartInfo using "cmd" as the program to be run,
                // and "/c " as the parameters.
                // Incidentally, /c tells cmd that we want it to execute the command that follows,
                // and then exit.
                System.Diagnostics.ProcessStartInfo procStartInfo =
                    new System.Diagnostics.ProcessStartInfo("cmd", "/c " + command);

                // The following commands are needed to redirect the standard output.
                // This means that it will be redirected to the Process.StandardOutput StreamReader.
                procStartInfo.RedirectStandardOutput = true;
                procStartInfo.UseShellExecute = false;
                // Do not create the black window.
                procStartInfo.CreateNoWindow = true;
                // Now we create a process, assign its ProcessStartInfo and start it
                System.Diagnostics.Process proc = new System.Diagnostics.Process();
                proc.StartInfo = procStartInfo;
                proc.Start();
                // Get the output into a string
                string result = proc.StandardOutput.ReadToEnd();
                // Display the command output.
                Console.WriteLine(result);
            }
            catch (Exception objException)
            {
                // Log the exception
            }
```

DLL File Example[3]

- Create C# Classes
- Generate DLL file
- Generate EXE file
- Run the EXE file

Create C# Classes

```
// File: Add.cs
namespace UtilityMethods {
    public class AddClass {
        public static long Add(long i, long j)
        {
            return (i + j);
        }
    }
}
```

```
// File: Mult.cs
namespace UtilityMethods {
    public class MultiplyClass{
        public static long Multiply(long x, long y) {
            return (x * y);
        }
    }
}
```

```
// File: TestCode.cs

using UtilityMethods;

class TestCode

{    static void Main(string[] args)  {

        System.Console.WriteLine("Calling methods from 
MathLibrary.DLL:");

        if (args.Length != 2){

            System.Console.WriteLine("Usage: TestCode 
<num1> <num2>");
return;}
        long num1 = long.Parse(args[0]);
        long num2 = long.Parse(args[1]);
}
 long sum = AddClass.Add(num1, num2);
        long product = MultiplyClass.Multiply(num1, num2);
System.Console.WriteLine("{0} + {1} = {2}", num1, num2,
sum);
        System.Console.WriteLine("{0} * {1} = {2}", num1,
num2, product);
    }
```

Generate DLL file and Generate EXE file

```
"C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc"          /target:library /out:MathLibrary.DLL Add.cs Mult.cs
```

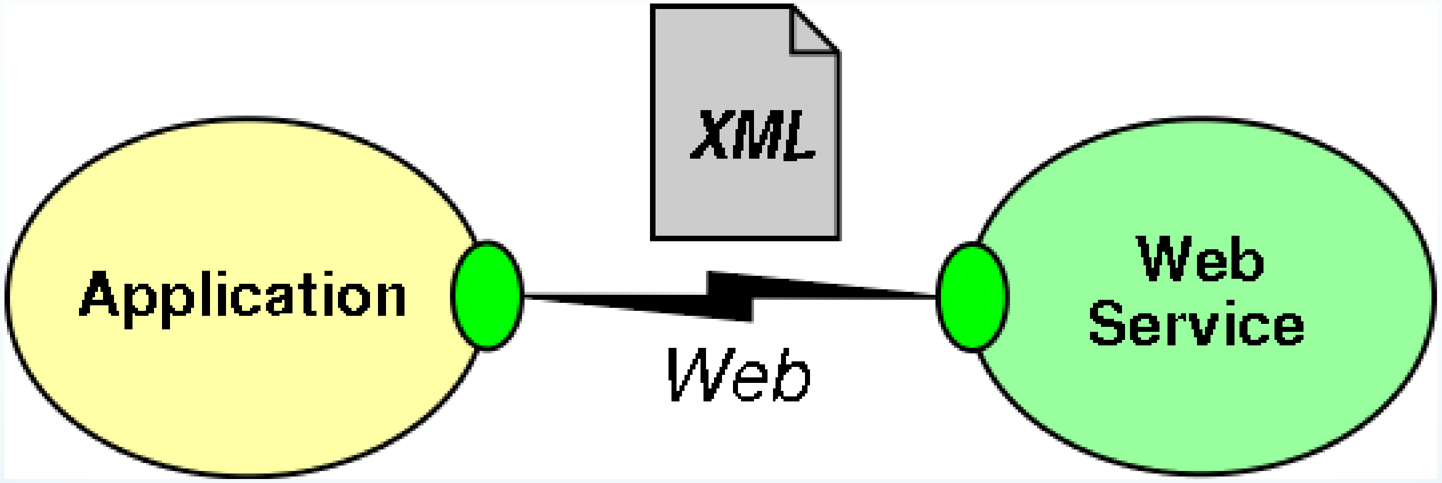
```
"C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc" /out:RunUtility.exe /reference:MathLibrary.DLL TestCode.cs
```

Run the utility:

```
RunUtility.exe 10 20
```

REST APIs

Web service
WSDL/REST



REST

- Representational State Transfer (REST) is an “architectural style” defined by Roy Fielding
  - The concepts of REST are independent of the Web, but the Web is well suited for the REST
- REST includes:
  - Resources(things) with
  - Unique ids (URLs) that can come in many
  - Representations(text, html, json, xml)
  - Verbs(GET, PUT, POST, DELETE)

REST

Most common operators

GET

Retrieve a representational of resource (without changing it)

PUT

Create or replace a resource by supplying representational to it

DELETE

Ensure that a given resource is no longer exist

POST

Augment a resource with additional representational

REST

Representational

As an external user you cannot manipulate a resource directly. Instead you manipulate representation of that resource

- Many people can “get ” representation of single resource
- Same resource can be manipulated in different ways

WEB and REST

Unique IDs for resources (URIs)

Verbs(HTTP operators)

Multiple representations (Media Types)

Why REST?

- The Web is an example of a REST system!
- All of those Web services that you have been using all these many years - book ordering services, search services, online dictionary services, etc - are REST-based Web services.

Restlet

Restlet is a Java framework for implementing REST architecture.

- Operators, Resources, Representations are all class entities in Restlet
- Highly pluggable implementation to support extensibility and interfaces to other web technologies

Atom,GWT, JSON,XML,SSL,Jetty, etc..

Calling several APIs

- Can be done programmatically
- Can use Business Process Management Framework
  - BPEL/BPMN
- Can use a workflow language
  - TAVERNA

Handling **input files** and **output files** REST-fully

Current Problem

Solution #01
Enable File upload via REST commands

Input:

- Upload entire files to the web service.
  - E.G. upload pictures to Facebook, or files to Dropbox
  - Granted there are UIs to facilitate this, and for this first deliverable there is no user interface.
- Achieved through standard HTTP request verbs
  - E.G. **POST** , **PUT**
  - Make clear API(s) using HTTP for file uploads.

Solution #01
Enable File upload via REST commands

Output:

- Teams have more flexibility in module output.
- **OPTION #01**
  - Return output as <u>JSON</u> response.
  - This is a very common return format for API calls in the real world.
- **OPTION #02**
  - Return <u>download links</u> to output files.
  - Links would be returned as part of a JSON response (as opposed to all the information being contained in a JSON response as with the first option).
- **SUGGESTION** Examine the responses from API calls from available services like Twitter

Solution #02
Create an executable and a Web Service

- This solution would require the development of **2 components**
  1. A desktop executable, and
  2. Web service (APIs w/ no UI)

- The executable has the responsibilities of:
  - Parsing input files (e.g. from command line)
  - Calling the APIs of associated web service
  - Handling the responses, and
  - Ultimately providing the user with the outputs

- The web service still does most of the heavy processing

Solution #02
Create an executable and a Web Service

OUTPUT:

- Just like with Solution #1 teams have more flexibility when it comes to outputs
- Executable affords more output options.
  - E.G. the executable could be in control of creating the output files.

Solution for SENG Workshops

- Both solutions are applicable to all three projects
- BUT they are **SUGGESTIONS!!!**
  - If you can, surprise us!
  - Creative, alternative and effective software designs are always impressive (to us).
- For deliverable 2 there will be points allocated to the adoption rate of your modules.
  - Practical indicator of design quality => How many people use it!
- Document Well
  - Your solution can't be used if no one knows how to use it *properly*!
- **ASK QUESTIONS!!!**
  - Filling in gaps in your knowledge and information provided : That's part of the **real process** out there
  - Asking effective questions early is paramount

Common Mistakes

- Component run accurately , but Log file incomplete or doesn’t exist
- No clear instructions on how to execute the component.
- The group said the version on their website is the wrong version, they will upload the correct version as soon as possible.
- Clear execution instructions, but lack of unit testing, errors generated when running the component
- Output doesn’t change when changing input parameters (i.e. hardcoded the parameters)
- Who is doing what in the group, clarify from the beginning don’t leave it to late.