

# CE0825a: Object Oriented Programming II

## 12: Documentation, Bugs, Versioning

James A Sutherland

Abertay University

Monday, 11th April 2016

# Documentation and Testing

- Generally ignored in 'programming' courses, but vital!
- Testing estimated between 10 and 50 percent of total time
- Documentation is not just for users, but yourself

# Documentation and Testing

- Generally ignored in 'programming' courses, but vital!
- Testing estimated between 10 and 50 percent of total time
- Documentation is not just for users, but yourself

# Documentation and Testing

- Generally ignored in 'programming' courses, but vital!
- Testing estimated between 10 and 50 percent of total time
- Documentation is not just for users, but yourself

# General Commenting

- Vital *aide memoire* for future maintainer — probably you!
- Meaningful variable, class, method names
- Single letter variables
- Not all comments are useful!

# General Commenting

- Vital *aide memoire* for future maintainer — probably you!
- Meaningful variable, class, method names
- Single letter variables
- Not all comments are useful!

# General Commenting

- Vital *aide memoire* for future maintainer — probably you!
- Meaningful variable, class, method names
- Single letter variables
- Not all comments are useful!

# General Commenting

- Vital *aide memoire* for future maintainer — probably you!
- Meaningful variable, class, method names
- Single letter variables
- Not all comments are useful!



# Useless Comments and Names

```
/* Print a string */  
void blah(String x)  
{  
    // Print the string:  
    System.out.println(x);  
}
```

# Useful Version

```

import java.util.*;
import javax.naming.directory.*;
public class lecture12_checkemail {
    /* Check if the provided email is valid, by looking for a mail
       server for it.
       */
    public static boolean isValidEmail(String email) {
        try {
            Properties env = new Properties();
            env.put("java.naming.factory.initial",
                "com.sun.jndi.dns.DnsContextFactory");
            final InitialDirContext idc = new
                InitialDirContext(env);
            String [] parts=email.split("@");
            Attributes attrs = idc.getAttributes(parts[1],
                new String []{"MX"});
            String mailservers =
                attrs.get("MX").get(0).toString();
            System.out.println(mailservers);
            return parts.length==2;
        } catch(Exception e) { System.err.println(e); return
            false; }
    }
    /* Test if all the email addresses on the command line are
       valid.
       */
    public static void main(String args[]) {
        for(String email : args) {
            if (!isValidEmail(email)) {

```

# Javadoc

- Java's own documentation is written via Javadoc
- `http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html`
- SWT too:  
`https://www.eclipse.org/swt/javadoc.php`
- Auto-generates HTML with cross-links
- Documents classes, fields, methods, interfaces, exceptions...

# Javadoc

- Java's own documentation is written via Javadoc
- <http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>
- SWT too:  
<https://www.eclipse.org/swt/javadoc.php>
- Auto-generates HTML with cross-links
- Documents classes, fields, methods, interfaces, exceptions...

# Javadoc

- Java's own documentation is written via Javadoc
- `http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html`
- SWT too:  
`https://www.eclipse.org/swt/javadoc.php`
- Auto-generates HTML with cross-links
- Documents classes, fields, methods, interfaces, exceptions...

# Javadoc

- Java's own documentation is written via Javadoc
- `http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html`
- SWT too:  
`https://www.eclipse.org/swt/javadoc.php`
- Auto-generates HTML with cross-links
- Documents classes, fields, methods, interfaces, exceptions...

# Javadoc

- Java's own documentation is written via Javadoc
- `http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html`
- SWT too:  
`https://www.eclipse.org/swt/javadoc.php`
- Auto-generates HTML with cross-links
- Documents classes, fields, methods, interfaces, exceptions...

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>



# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>



# Javadoc tags

Special annotations to help the HTML generation<sup>1</sup>

`author` Who wrote it

`@deprecated` 1.0

`@exception` 1.0

`{@link}` 1.2

`@param` 1.0

`@return` 1.0

`@see` 1.0

`@since` 1.1

`@throws` 1.2

`@version` 1.0

---

<sup>1</sup><http://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>

# Version Control

- Share code
- Who changed something?
- What?
- When?
- Why?

# Version Control

- Share code
- Who changed something?
- What?
- When?
- Why?

# Version Control

- Share code
- Who changed something?
- What?
- When?
- Why?

# Version Control

- Share code
- Who changed something?
- What?
- When?
- Why?

# Version Control

- Share code
- Who changed something?
- What?
- When?
- Why?

# Version Control Examples

- Publicly visible in open source projects
- Github — free student pack

# Version Control Examples

- Publicly visible in open source projects
- Github — free student pack



# Bugs

- Inevitable part of software development
- Gather information from users — manually or not
- Crash reporting, field telemetry
- Keep track of known problems: document, fix, monitor

# Bugs

- Inevitable part of software development
- Gather information from users — manually or not
- Crash reporting, field telemetry
- Keep track of known problems: document, fix, monitor

# Bugs

- Inevitable part of software development
- Gather information from users — manually or not
- Crash reporting, field telemetry
- Keep track of known problems: document, fix, monitor

# Bugs

- Inevitable part of software development
- Gather information from users — manually or not
- Crash reporting, field telemetry
- Keep track of known problems: document, fix, monitor

# Lab Task Week 12

- 1 Complete and submit labs by **tonight**
- 2 Get coursework ready for **next week**
- 3 Note: I will be marking third year in 4506 today

# Lab Task Week 12

- 1 Complete and submit labs by **tonight**
- 2 Get coursework ready for **next week**
- 3 Note: I will be marking third year in 4506 today

# Lab Task Week 12

- 1 Complete and submit labs by **tonight**
- 2 Get coursework ready for **next week**
- 3 Note: I will be marking third year in 4506 today