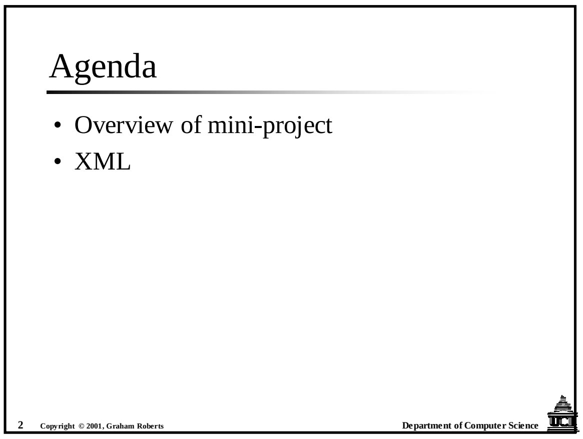
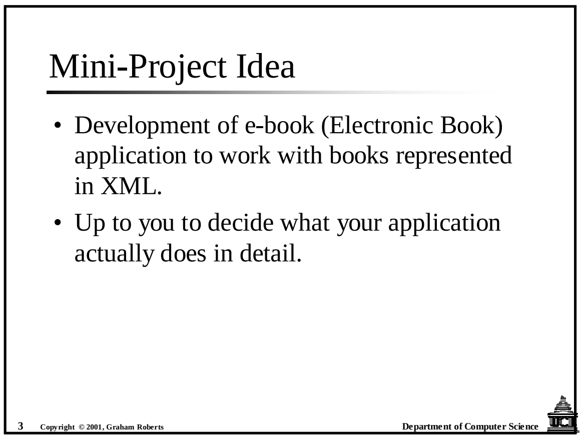
2B11

Mini-Project I



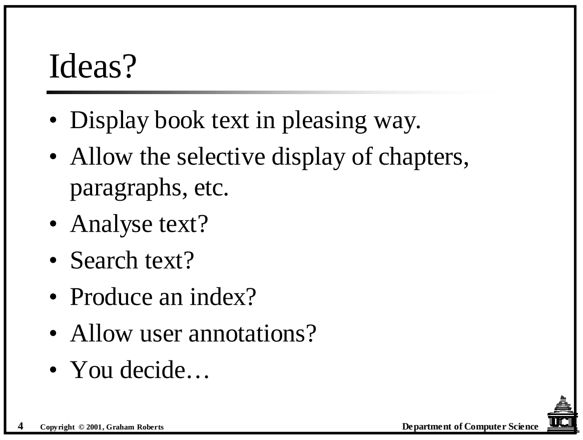
Agenda

- Overview of mini-project
- XML



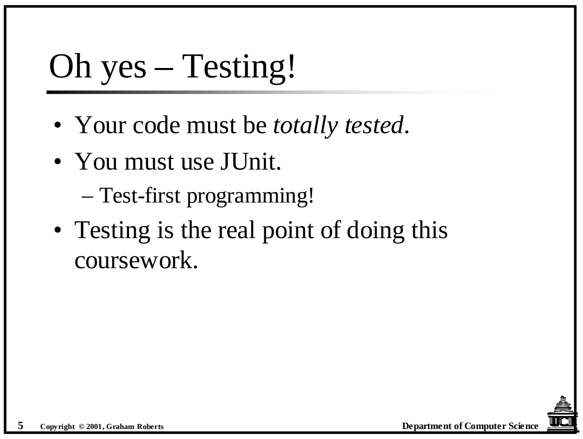
Mini-Project Idea

- Development of e-book (Electronic Book) application to work with books represented in XML.
- Up to you to decide what your application actually does in detail.



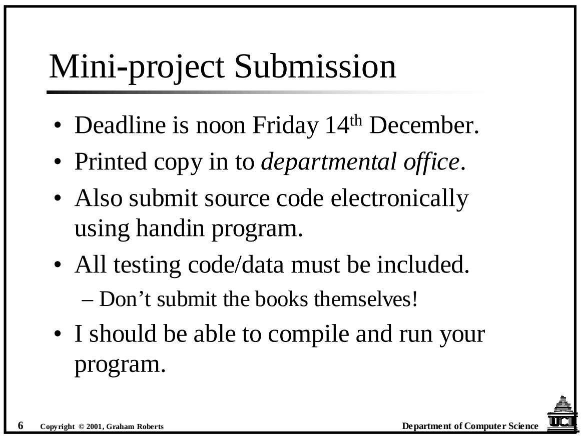
Ideas?

- Display book text in pleasing way.
- Allow the selective display of chapters, paragraphs, etc.
- Analyse text?
- Search text?
- Produce an index?
- Allow user annotations?
- You decide…



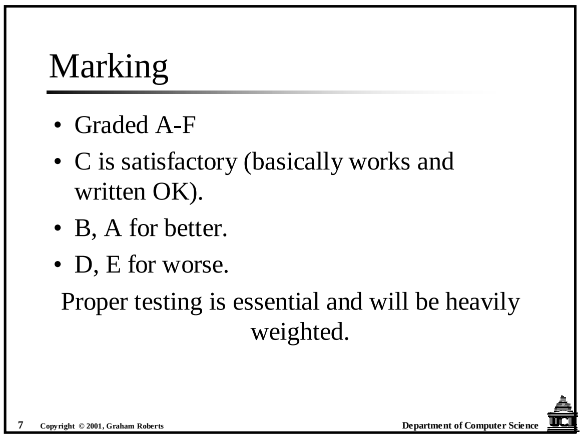
Oh yes – Testing!

- Your code must be *totally tested*.
- You must use JUnit.
  - Test-first programming!
- Testing is the real point of doing this coursework.



Mini-project Submission

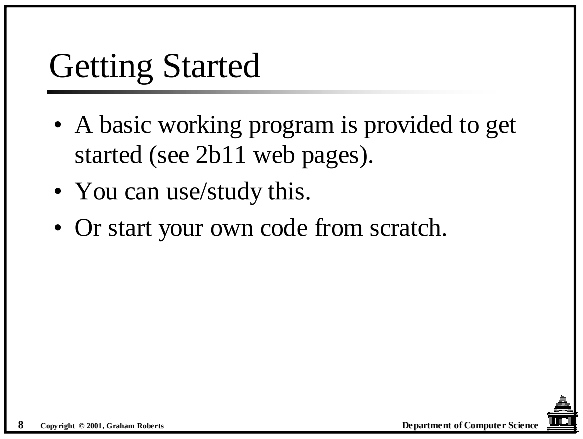
- Deadline is noon Friday 14th December.
- Printed copy in to *departmental office*.
- Also submit source code electronically using handin program.
- All testing code/data must be included.
  - Don’t submit the books themselves!
- I should be able to compile and run your program.



Marking

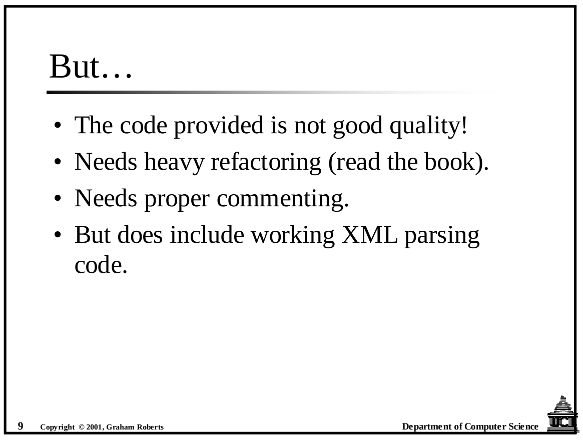
- Graded A-F
- C is satisfactory (basically works and written OK).
- B, A for better.
- D, E for worse.

Proper testing is essential and will be heavily weighted.



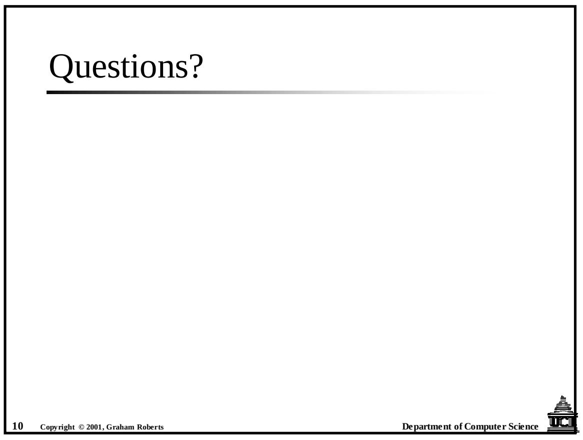
Getting Started

- A basic working program is provided to get started (see 2b11 web pages).
- You can use/study this.
- Or start your own code from scratch.

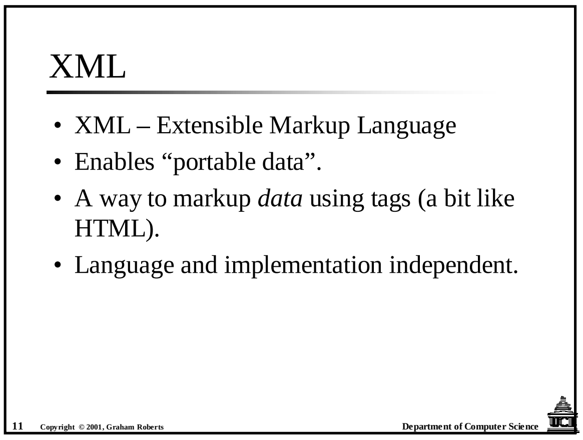


But…

- The code provided is not good quality!
- Needs heavy refactoring (read the book).
- Needs proper commenting.
- But does include working XML parsing code.

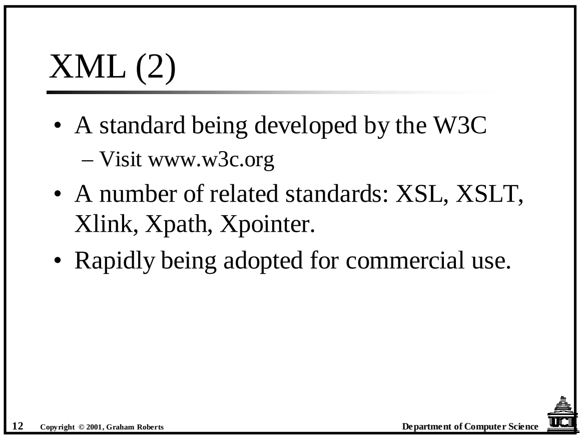


Questions?



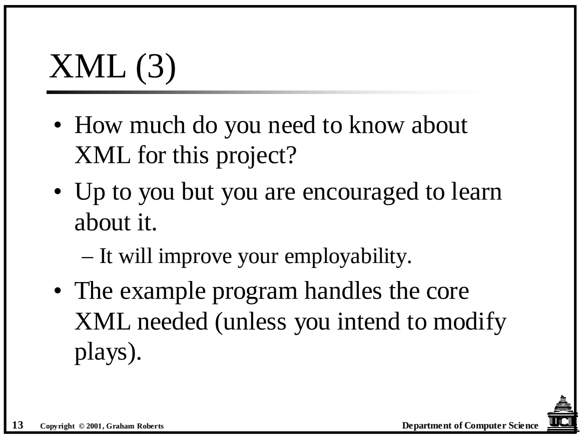
XML

- XML – Extensible Markup Language
- Enables “portable data”.
- A way to markup *data* using tags (a bit like HTML).
- Language and implementation independent.



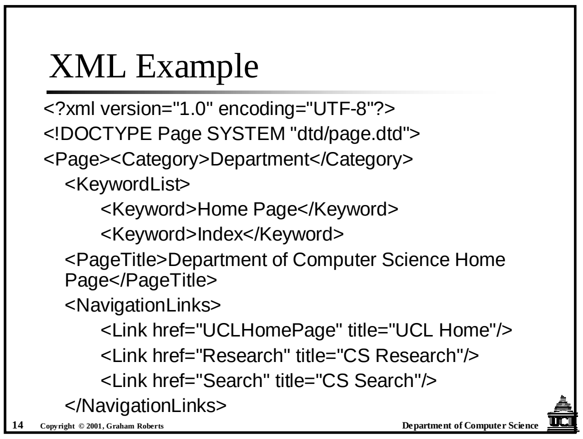
XML (2)

- A standard being developed by the W3C
  - Visit www.w3c.org
- A number of related standards: XSL, XSLT, Xlink, Xpath, Xpointer.
- Rapidly being adopted for commercial use.



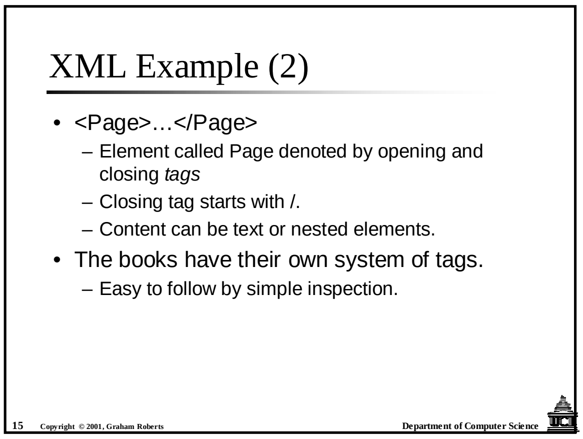
XML (3)

- How much do you need to know about XML for this project?
- Up to you but you are encouraged to learn about it.
  - It will improve your employability.
- The example program handles the core XML needed (unless you intend to modify plays).



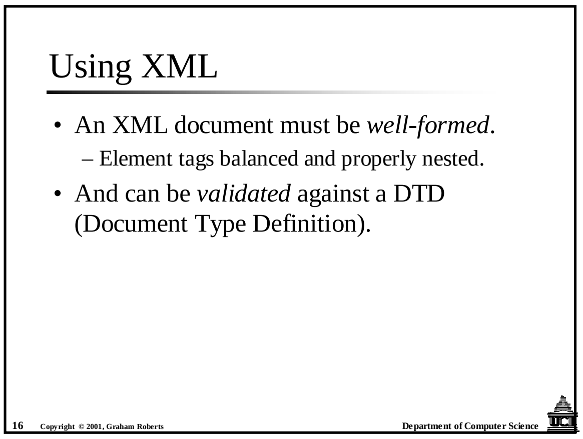
XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Page SYSTEM "dtd/page.dtd">
<Page><Category>Department</Category>
  <KeywordList>
      <Keyword>Home Page</Keyword>
      <Keyword>Index</Keyword>
  <PageTitle>Department of Computer Science Home
  Page</PageTitle>
  <NavigationLinks>
      <Link href="UCLHomePage" title="UCL Home"/>
      <Link href="Research" title="CS Research"/>
      <Link href="Search" title="CS Search"/>
  </NavigationLinks>
```



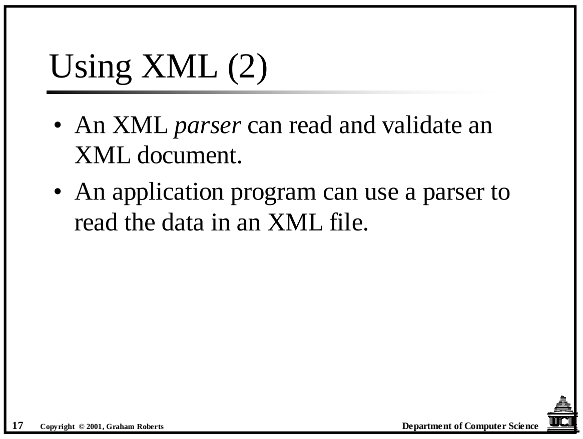
XML Example (2)

- <Page>…</Page>
  - Element called Page denoted by opening and closing *tags*
  - Closing tag starts with /.
  - Content can be text or nested elements.
- The books have their own system of tags.
  - Easy to follow by simple inspection.



Using XML

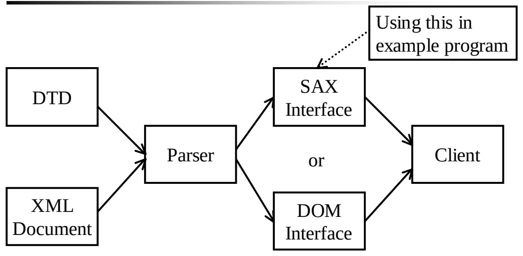
- An XML document must be *well-formed*.
  - Element tags balanced and properly nested.
- And can be *validated* against a DTD (Document Type Definition).



Using XML (2)

- An XML *parser* can read and validate an XML document.
- An application program can use a parser to read the data in an XML file.

Using XML (3)



Using XML (3) – SAX

- Simple Api for Xml.
- Works by using call-backs method as each element tag or content is encountered.
- Client can provide call back method implementations to store data and build a data structure.

Using XML(5)

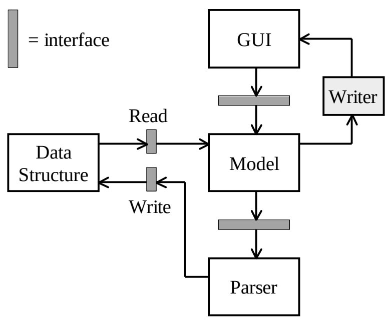
- The example code uses a SAX2.0 parser implementation.
  - – Use the Apache Xerces parser.
- The xerces.jar files need to be in your classpath.
- See the 2b11 web pages for details.
- Read comments in code.

Questions?

The Example Program

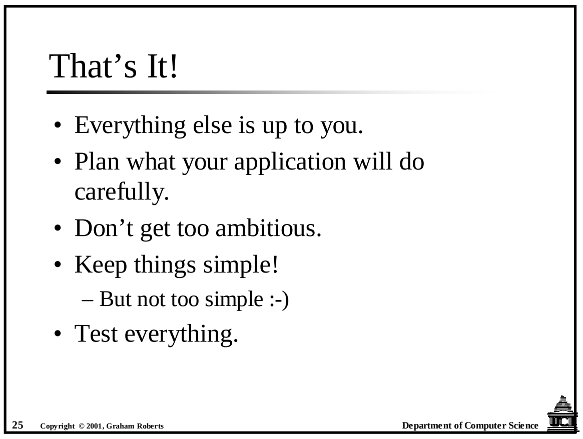
- Can parse a book and build a rudimentary data structure.
- No testing code included – you have to write that!

Example Program (2)



Example Program (3)

- Overall structure broken into 4 main components.
  - – 2 packages.
- Each component will be implemented using some number of classes.
- Java interfaces and a connector class define the connections between each component.
- Work to interfaces rather than specific classes.



That's It!

- Everything else is up to you.
- Plan what your application will do carefully.
- Don't get too ambitious.
- Keep things simple!
  - But not too simple :-)
- Test everything.