

COMPSCI 732 FC 2007

Homework: XML Document and DTD Construction

5 April 2007

For this homework you will look at the development of a DTD for a particular set of information and then create an XML document to hold information which conforms to the DTD.

Course descriptions at a university have a particular form. Look at the examples below and create a DTD for courses within a department. The top level element should be called departmentSchedule and should contain the department's name, the faculty it belongs to, the year this schedule of courses is for (2006) as well as details for each course. Note that the ordering of course information (as in the examples) is important here.

Engineering Design 2 **SOFTENG 206 DC**

Schedule

Semesters 1 and 2

Assessment

60% projects; 40% tests

Organisation

2 lectures and 2 hour lab per week

Description

This course provides experience for project work. Students work on projects over two semesters, and will be exposed to a number of topics relating to the systematic production of software.

Contents

Programming craft (unit testing, debugging, test-first programming), personal software process, software requirements specification, extreme programming

Software Engineering 1 **SOFTENG 251 SC**

Schedule

Second semester

Assessment

20% assignment; 20% test; 60% exam; separate pass required for practical and theory

2002 Text

Fowler, M "UML Distilled" Addison-Wesley 1999 (??)

Organisation

3 lectures and 1 tutorial per week

Description

This paper looks in the first half of the course at extending the "programming in the small" concepts and practices from SOFTENG 250. The Unified Modelling Language is introduced as a program design notation, and object-oriented programming concepts like inheritance are introduced. Various

supporting ideas like coding standards, refactoring, unit testing, debugging and program documentation are learned. An introduction to design patterns is given. In the second half of the course "programming in the medium" concepts and skills are developed. These include the use of more UML constructs and processes for life cycle phases of requirements, specification and design, and integration and system testing. Use databases and basic Java web facilities to build distributed, on-line software systems. System documentation, version control and peer review are also introduced.

Contents

UML, Inheritance, Coding standards, Refactoring, Unit testing, Coverage Debugging, Documentation, Design Patterns Systems development life cycle, User Interface Design Integration and System testing, Version control techniques, Databases and SQL, JSP and Servlets

Algorithms

SOFTENG 253 SC

Schedule

Second semester

Assessment

Assignments 25%; Test 10%; Exam 65%; must pass both theory and practical

2002 Text

Collins, William J., Data Structures and the Java Collections Framework, McGraw Hill 2002

Related Reading

Data Structures, Algorithms, and Applications in Java by Sartaj Sahni, McGraw-Hill

Organisation

3 lectures per week, 1 two hour lab per week

Description

To provide the student with an understanding of advanced algorithm design and the analytical and empirical behaviour of advanced algorithms and data structures.

Contents

Using Java Collections, Sorting, File Structures and Indexing, Graph Algorithms, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

Now create a XML document which matches the DTD developed above and ensure that this loads in a parser (e.g., Internet Explorer).