Due on: Thursday 28.4.2005 by 14:00 (in class)

Exercise 3

a) Is the following XML document well-formed? Describe and correct all errors.

```
<?xml version="1.0" ?>
<cds>
    <cd 2sided="true">
        <title>Grace Live </title>
        <artist id="1">Jeff Berry Band </artist>
        <TRACKS>
            <track>Take me higher</track>
            ...
            <track>If you say go</track>
        </tracks>
    </cd cat="rock">
    <cd 2sided="false">
        <title>Winter Pays For Summer </title>
        <artist id=’2">Glen Phillips </artist>
        <tracks>
            <track>    Duck And Cover </track>
            ...
            <track> Don't Need Anything </track>
        </tracks>
    </cd>
    <cd cat="country"/>
</cds>
```

b) Model the following information as XML document.

Name	Email	Phone	Affiliation
John Doe	john@doe.com	0621 / 2040503	Doe Inc.
Jane Doe	jane@doe.com	0621 / 2040502	Doe Inc.
Kelly Edwards	Kelly@abc.com	+85 544 / 1211245	MaryLand Inc.

Company	Customer since	Customer until
Doe Inc	10/2001	2/2005
MaryLand Inc.	2/2002	---

Exercise 4

a)

```
From: condolezza@rice.com
To: george@bush.com, colin@powell.com
Cc: anan@kofi.com, toni@blair.com
Subject:  Spam mails

Arafat is spamming your email, please don't answer him.

[Attachment: "Your outlook.jpg"; image/jpeg; Contents]
[Attachment: "Spam mails for dummys.pdf"; application/pdf; Contents]
```

1. Write a DTD for emails that allows to represent any information in the example above.
2. Extend the DTD so that the body text may contain markup style: bold, italic and underlined.
3. Is this a document-centric or data-centric DTD?

b)

```
public class Currency extends BigDecimal
{
      private double value;

      public Currency(double v) {
             value = v;
          }
      public double getValue() {return value;}

      public Currency add(Currency val) {
             return new Currency(val+val);
          }
      public Currency multiply(double val){
             return multiply(new Currency(val));
          }
      public Currency divide(double val)throws ArithmeticException{
             return divide(new Currency((double)val));
          }
}
```

1. Write a DTD that allows to represent Java classes as the one given above including at least the class definitions, inheritance and interface implementation, member variables, and public and private methods including argument and return types (but not the actual method implementations).

2. Write an XML document that represents the given example and is valid w.r.t. your DTD in 1).