UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1998

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BSc Honours Degree in Mathematics and Computer Science Part II
MSci Honours Degree in Mathematics and Computer Science Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Associateship of the Royal College of Science
Associateship of the City and Guilds of London Institute

PAPER 2.7 / MC2.7

SOFTWARE DESIGN II

Tuesday, April 28th 1998, 4.00 - 5.30

Answer THREE questions

For admin. only: paper contains 4 questions

Section A *Use a separate answer book for this section*

1a As an expert, you are asked to make a formative heuristic evaluation of a new prototype control interface for a video recorder. Provide criteria by which you would assess the interface, and indicate why these criteria assess usability.

b A protoype task-oriented design of a dialogue box intended for use by expert dealers in on-line negotiation is criticised for obscuring critical information on a dense screen display with many open stock transactions.

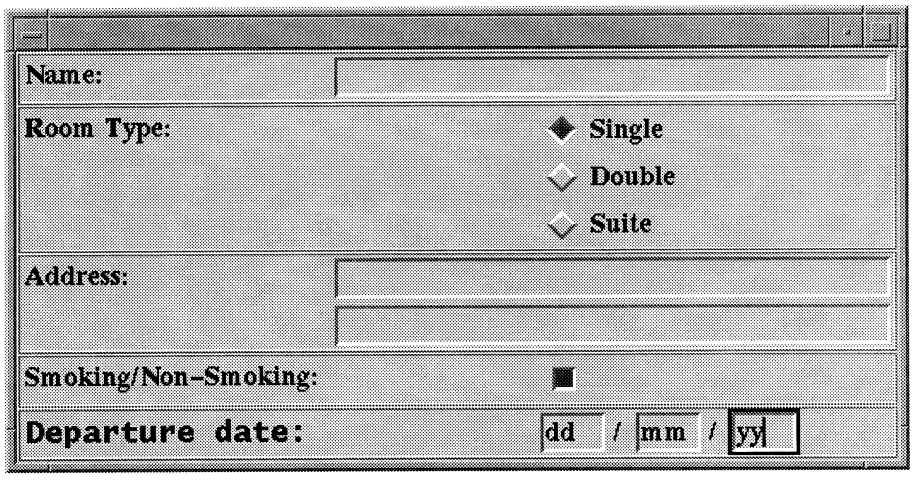
i) What human processing capacity has the task oriented dialogue ignored?
ii) Indicate how you might expect to achieve a design which is more acceptable to the users?
iii) As a professional in the field, what further evaluation processes would you recommend once the design seems acceptable to the users?

c Briefly describe:

i) the processes of user centred design,
ii) the benefits claimed for this design method, and
iii) the chief difficulties and problems entailed.

Parts a, b and c of the above question carry, respectively, 30%, 30% and 40% of the total marks.

2 Imagine a hotel where a guest may register with an Automated Reception, by filling in a form like the following on a computer screen.



a State five distinct guidelines which are broken by the above form, then briefly discuss how to proceed with a re-design intended to remedy these and other deficiencies.

b You are asked to design a supplementary form on which a guest can indicate requirements for wake-up time and method (*alarm*, *radio*, *TV*, or *telephone*), and for breakfast time and content (*juice, cereal*, *toast*, *boiled egg*, *tea*, etc.).
i) Provide a simple hierarchical task diagram for the joint task of displaying options, completing, and submitting the new form. Clearly indicate the the information you are trying to display and collect for each elementary task.
iii) Sketch a suitable form, indicating informally the type of widgets and manipulative actions you include to match your elementary tasks.

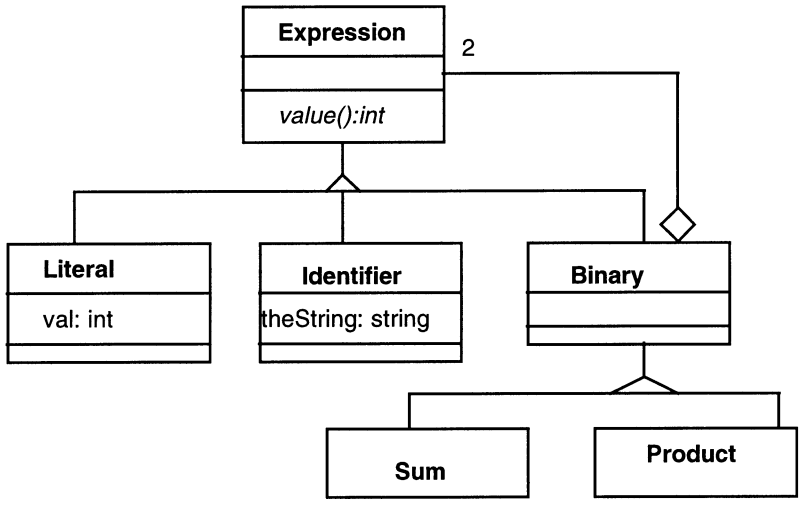
Parts a, b(i) and b(ii) of the above question carry, respectively, 30%, 40% and 30% of the total marks.

Section B (*Use a separate answer book for this Section*)

3 Consider constant expressions, like `3*(4+2)+anIndent`: A constant expression is an integer literal (e.g. `22`), or an identifier (e.g. `anIdent`), or the sum of two constant subexpressions (e.g. `22+anIdent`), or the product of two constant subexpressions (e.g. `33*(22+anIdent)`).

The value of an integer literal is the literal itself. The value of an identifier is found by looking it up in a table. The value of a sum is calculated by adding the values of the two subexpressions. The value of a product is calculated by multiplying the values of the two subexpressions.

The following OMT object model class diagram describes constant expressions:



A summary of the OMT notation can be found on page 4.

a Write Java classes that support the creation of constant expressions and the calculation of their values as described above.

For the lookup of the values of identifiers use the following class:

```
class table{
      public int lookUpValue(String ident){. . .}
      public table(){. . .}
      public storeValue(String ident, int val){. . .}
      . . . };
```

b Write a test function that

i) creates a lookup table,

ii) stores in the lookup table the value `2` for the identifier `anIdent2`, and the value `3` for the identifier `anIdent3`,

iii) creates the expression `3*(anIdent2+anIdent3)` and evaluates it.

The two parts carry, respectively, 70% and 30% of the marks.

Turn over

4 Consider the following simplified description of bank accounts:

There are several banks holding accounts owned by people. Every bank has a name and a daily interest rate. Daily interest rates may change.

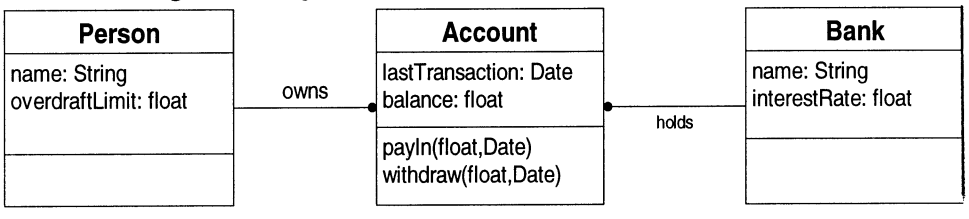
Every person has a name, and may own more than one account. Every person has an overdraft limit, which sets a limit to the amount this person may be overdrawn on any of his/her accounts. Note that the overdraft limit does not depend on banks or accounts.

Money may be paid into or withdrawn from an account.

When a amount is paid into an account on a given date, interest is calculated by adding to the balance the product of the number of days from the last transaction until the current transaction, and the interest rate of the holding bank. Then, the balance is incremented by the amount.

When an amount is withdrawn from an account on a given date, interest is calculated by adding to the balance the product of the number of days from the last transaction until the current transaction, and the interest rate of the holding bank. Then, the balance is decreased by the amount, provided that it does not fall below the owner's overdraft limit; otherwise, an error message is printed.

The following OMT object model class diagram describes bank accounts:



a Write Java classes to support the above.

For the representation of dates use a class `Date` defined as follows:

```
class Date{
    Date(int day, int month, int year) { . . . }
    int daysUntil(Date aDate){ . . . }
    // returns number of days between receiver and aDate
    . . . };
```

b Write a test function that:

i) creates two banks: the Listening Bank with an interest rate of 0.030, and the Talking Bank with an interest rate of 0.025,

ii) creates two people: Kenneth with an overdraft limit of 100.00, and Gordon with an overdraft limit of 45,000.00,

iii) creates A1, an account for Kenneth, with the Listening Bank, on 2nd March 1995, with an initial balance of 230.00, and creates A2, an account for Gordon, with the Talking Bank, on 5th May 1996, with an initial balance of 56,000.00,

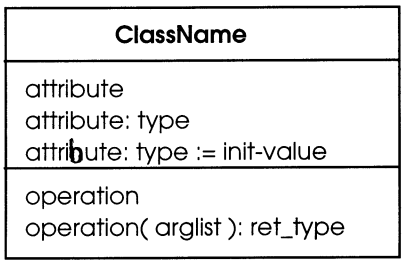
iv) withdraws 300.00 from A1 on the 10th January 1997, and pays 500.00 into A2 on the 4th March 1998,

v) changes the interest rate of the Talking Bank to 0.035.

The two parts carry, respectively, 70% and 30% of the marks.

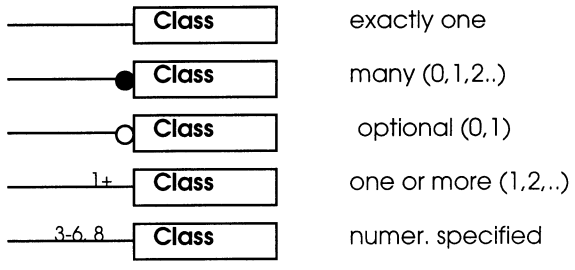
OMT: Basic Notation for Object Models

Class:

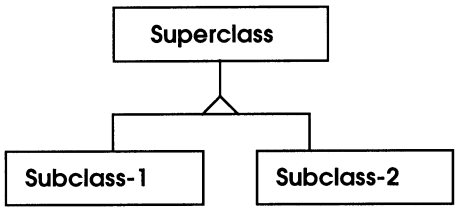


Association

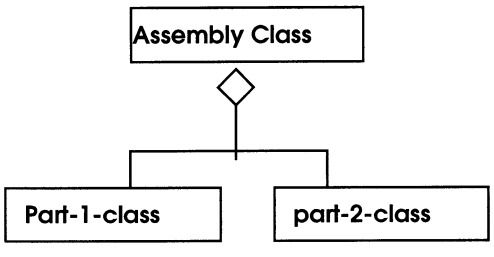
Multiplicity of Association/Aggregation



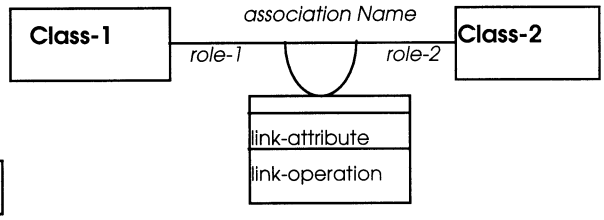
Generalization (Inheritance)



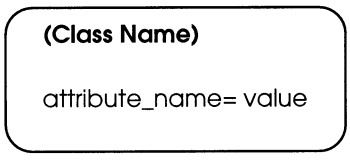
Aggregation



Link Attributes

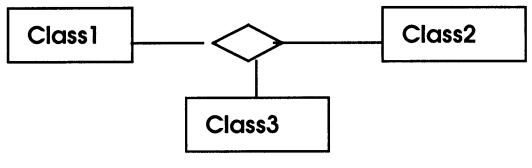


Object Instance



$ for class operations/attributes

Ternary Association



End of Paper