UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1996

BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
BSc Honours Degree in Mathematics and Computer Science Part III
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 City and Guilds of London Institute
Associateship of the Royal College of Science

PAPER I3.4 / 3.51

OBJECT ORIENTED DESIGN AND PROGRAMMING

Tuesday, April 30th 1996, 1.30 - 3.30

Answer THREE questions

For admin. only: paper contains
4 questions
4 pages (excluding cover page)

1 Consider interest paying accounts, where interest is added to the balance on each transaction, and it is calculated according to the following formula:

> Balance on the day of latest transaction * 0.0003 * Interest Rate * (Number of days between Current Transaction and Previous Transaction)

We distinguish `Golden Accounts` and `Privileged Accounts`: The interest rate for all `Golden Accounts` is set centrally, whereas for `Privileged Accounts` the interest rate is agreed upon on account creation.

a Write C++ classes (interface and implementations) to describe both kinds of accounts.

For `Golden Accounts` provide the possibility to:

i) set the interest rate for all accounts to a new value;

ii) create a new account on a certain date with an initial balance;

iii) deposit an amount of money on a certain date: (calculate and add interest, and then increase the balance);

iv) withdraw an amount of money on a certain date: (calculate interest, decrease the balance);

v) enquire the balance of the account.

For `Privileged Accounts` provide the possibility to:

i) create a new account on a certain date, with an interest rate and with an initial balance;

ii) deposit an amount of money on a certain date: calculate interest, and then increase the balance;

iii) withdraw an amount of money: calculate interest, and then decrease the balance;

iv) enquire the balance of the account.

b Write a main program in which

i) the interest rate for `Golden Accounts` is set to 7.0;

ii) the variable GA contains a `Golden Account` created on the 10th February 1993, with 140.00 initial balance;

iii) the variable `PA1` contains a `Privileged Account` created on the 25th February 1993, with 100.00 initial balance, and interest rate of 6.5;

iv) the variable `PA2` contains a `Privileged Account` created on the 17th March 1993, with 80.00 initial balance, and interest rate of 7.5;

v) pay into GA the amount of 60.00 on the 16th June 1994;

vi) the interest rate for `Golden Accounts` is set to 6.0.

The two parts carry, respectively, 80% and 20% of the marks.

Note: For the description of dates use the following definitions:

```
enum month {    jan, feb, mar, apr, may, jun,
                jul, aug, sep, oct, nov, dec};
class Date { public:
    Date(int d=1; month m=jan; int y=1996);
    int operator -(Date);
    // returns number of days between receiver and argument
...};
```

2 The following describes part of an airline's database:

i) A flight has a code number, a date, and a price. It takes off at a certain city and lands at another city.

ii) A passenger has a name and a address. A passenger may be booked for a flight. There are firm bookings, where the complete price has been paid, and flexible bookings, where only part of the price has been paid.

iii) Every flight is serviced by two pilots and six crew members.

iv) A pilot has a name, an address and an identification number.

v) A crew member has a name, an address and an identification number.

vi) Information about a flight may be printed: The name, address and identification number of the pilots will be printed, followed by the name, address and identification number of the crew members, followed by the name, address and remaining price to be paid (if any) for all the passengers.

a Draw an OMT class diagram describing the above. Indicate the type of the attributes, and the types of the arguments and results of the operations.

Note: Not *all* aspects of the above situation can be expressed in the OMT class diagram.

b Draw an OMT object diagram where we have Mr. Jones, of 13 Leicester Square, booked firmly on the 164 flight, costing £925 for the 25 March 96, from Athens to Caracas. Mr. Jones is also booked flexibly on the 278 flight, costing £260 for the 25 April 96, from Caracas to Lima, and he has paid £220. Mrs. Smith, from 180 Queen's Gate, is booked firmly on the 164 flight, costing £925 for the 10 June 96, from Athens to Caracas.

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

Note: On the last page you can find a summary of the OMT notation for object models.

Turn over ...

3a Explain the use of *clustering* in Harel's State Charts.

b A central heating system is controlled by an ON/OFF switch and a room thermostat. When the switch is ON and the room temperature t is less than temperature T° (the thermostat set point) , then the central heating boiler is turned on. When the temperature $t \geq T°$, then the boiler is turned off. When the control switch is OFF, the boiler is off independently of the room temperature t.

i) List the events which must be dealt with by the controller of the central heating system and the actions it must perform.

ii) Draw a State Chart to define the behaviour of the central heating controller.

The two parts carry, respectively, 20%, 80% of the marks.

4a Define a template class `Pair` which constructs pair objects from values of two types `T1` and `T2`. As an example of the use of `Pair`, give the definition of a `Pair` variable containing the string `"One"` and the integer value `1`.

b Given the following classes for `List` and `ListIterator`:

```
template <class Item>
class List  {
public:
    .....    }

template <class Item>
class ListIterator {
public:
    ListIterator(List<Item>& s);
    void first();   // position Iterator at start of list
    void next();    //position Iterator at next item
    bool isDone();  //true if finished traversing the List
    Item current() const;  // value of Item in List at current position
    }
```

give the C++ code for a template function `find` which takes as its parameters a `List` of `Pairs` of types `T1` and `T2`, and a value of type `T1` and returns the first `Pair` with this value of `T1`. Assume that the list always contains the value and that the operation == is defined for `T1`.

c A navigation device maintains a database of `Waypoints`, where a `Waypoint` is a named `Location`. A `Location` is specified by a latitude and longitude which are stored as floating point numbers (`float`).

i) Define types for `Waypoints` and `Locations`.

ii) Define a variable to hold the database.

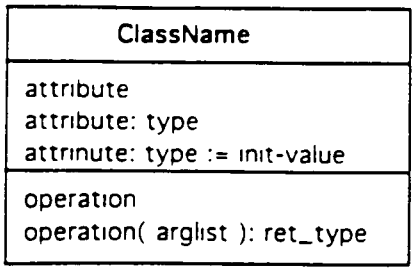
iii) Give a code fragment which shows how waypoints are retrieved from the database by name.

(Hint: use the templates from parts a & b).

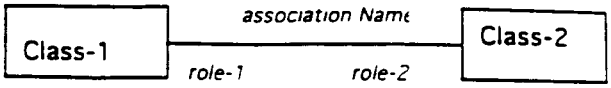
The three parts carry, respectively, 30%, 40% , 30% of the marks.

OMT: Basic Notation for Object Models

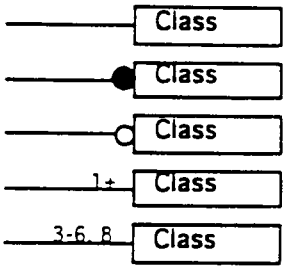
Class:



Association



Multiplicity of Association/Aggregation



exactly one

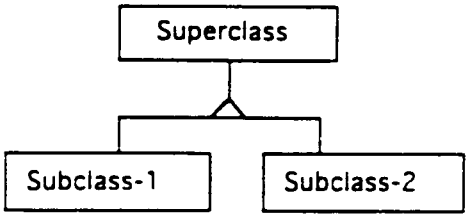
many (0,1,2..)

optional (0,1)

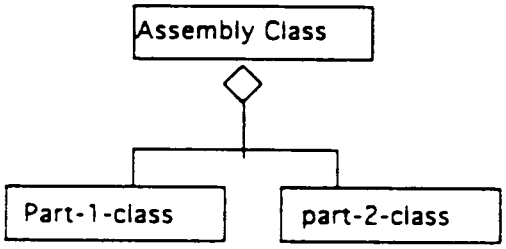
one or more (1,2,..)

numer. specified

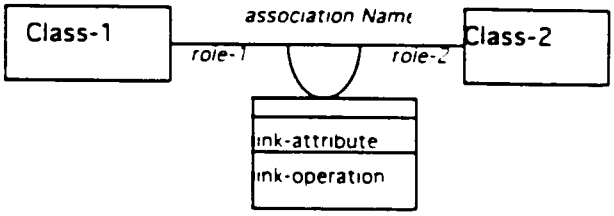
Generalization (Inheritance)



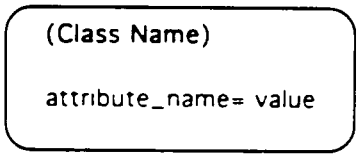
Aggregation



Link Attributes

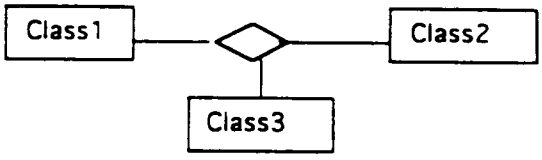


Object Instance



$ for class operations/attributes

Ternary Association



End of paper