A1. Indicate whether the following statements are true or false. For two (2) of your choices, you may provide an explanation of your choice of true or false.

**True**
- In UML, objects are used to model the application domain and can therefore model both the human activity and the application software.
- Object Oriented design results in low coupling between objects since it is not necessary for an object to know anything about the way in which another object will deliver the service.
- Encapsulation makes it possible to change internal logic or the way data is stored within an object and, if the signature is not changed, there is no other part of the system that needs to be changed.

**False**
- Use cases are definitions of internal processes within the system and are used as program specification.
- In use case diagrams, actors can be people, departments, other systems and devices.
- The management of all transfers across subsystem boundaries is handled by the interface classes.
- The current state of an object is fully described by the instance values of its attributes.
- If we show the same object in more than one sequence or collaboration diagram, then the object’s class will include all operations and attributes needed for these diagrams.
- Activity diagrams can be used to express the logic of an operation.
- Abstract operations can be used in superclasses and subclasses.

For the 6th item is could be argued that a boundary class is sometimes called an interface class. So this is a bonus mark because either answer was accepted.
A2. Referring to the diagram below, explain polymorphism.

Polymorphism, according to the text, is the “ability of different methods to implement the same operation and thus respond to the same message in different ways that are appropriate to their class”.

Your answer had to refer to the operations in this diagram and point out the different methods and responses to the same message.

Answers that were merely definitions or explanations that would apply to any type of class (i.e. that were not specific to polymorphism) got partial, if any, marks.

A3. An object may respond differently to the same message at different times.

Explain why, using an example.

For full marks, you had to mention object states. In the question, the “same message” means the same signature and associated parameters! (Indicating that with different parameters you get different responses is a trivial answer.)

One or two students got full marks by giving convincing arguments about asynchronous messages in real-time systems, but most good examples were based on more ordinary systems discussing the states of bank accounts, library books, etc.

A4. For systems with the characteristics described below, indicate whether object-oriented design is suitable (appropriate) or not.

management information system (mostly queries) not suitable

event-driven systems (e.g. aircraft navigation/control) suitable

multimedia data handling (presentations) suitable

scientific calculations (e.g. satellite orbit calculation) not suitable

One student added a comment that structured systems were outdated. I did not deduct marks, though I was tempted. O-O is not a silver bullet.
A5. Completing class diagrams is an iterative process. Sometimes you have to remove some classes because they are unsuitable for your system. For example, a class might be beyond the scope of your system. Give two other reasons for removing classes from your diagram. Then give an example for one (1) of your reasons.

Good answers would have used terms/phrases such as:
- they refer to the system as a whole – duplicate other class(es) – too vague – too specific – too tied up with physical input/output – is really an attribute, operation or association
And, if well explained, the following were also good answers:
- improvements related to coupling or cohesion – functional requirements changes
The examples were sometimes weak and did not precisely demonstrate the reason.

A6. Match the terms with the examples.

composition ___B___ A. a course may be offered several times
aggregation ___A___ B. a student has academic records
synchronous message ___B___ A. alarm system detects fire
asynchronous message ___A___ B. a customer record is needed
return ___A___ A. a requested record has been retrieved
call back ___B___ B. a specified time interval has elapsed

A7. Class diagrams evolve throughout the analysis and design phases in a project. What changes or refinements could you make to your class diagram as a result of completing the user interface design (reports and screen contents and layouts)? Name two changes/refinements and give an example for each.

In general, any of the following may need to be done after screens and reports have been designed:
- add, change, delete, combine:
  - boundary or control classes
  - operations, visibilities, attributes, associations...
- because:
  - client has requested changes to functionality, interface, performance, etc.
  - designer sees opportunities for correction or improvement based on interface requirements
Your answer would be more specific of course.
If you gave general reasons for changes or refinements that were not specific to user interface, you only got part marks.
By the way, you do not redesign your entity classes to match screen or report contents! You have to have better reasons for fixing or improving entity classes.
A8. What documentation would you provide for use cases?
   Name three (3) items of information in addition to the name of the use case. [3]

Good answers would have used terms/phrases such as:
- pre-conditions – post conditions – purpose – description – related objects/classes
- basic and alternate flows – multiplicity – actors – relation to other use cases
Anything else received part marks if it was plausible.
Some general answers such as “system boundary” were wrong.

A9. Define multiple inheritance using the following class diagram as an example. [3]

```
Vehicle
  ▲
LandVehicle  WaterVehicle
  ▲
Car  AmphibiousVehicle  Boat
```

Multiple inheritance means that a class has more than one superclass. You had to state clearly that
AmphibiousVehicle is the class exhibiting multiple inheritance. The other classes do not. Just
defining inheritance in general got low marks.

A10. When installing a system, you have choices of methods. Under what circumstances or situations
would you consider a pilot method to be preferred over a parallel method? Describe two (2)
circumstances or situations. [4]

Your two answers had to be different, not variations on the same theme.
If your answer included any of the following, you only got marks if you explained further because
these answers could apply to several implementation methods:
- simplicity of operation
- the system is small
- the new system is a package
Some answers that got marks:
- parallel may be too expensive (money and/or time)
- the system was decentralized and therefore appropriate for pilot
- users were not committed to new system and might not run parallel properly
- the old and new system are difficult/impossible to compare
- the new system may need changes because:
  - the system is an evolutionary prototype
  - practical experience is needed to prove the system
A11. For each of the following non-functional requirements, describe two (2) aspects or features of the system’s architecture that would be necessary in order to fulfill that requirement.

Some good answers (there are more!)

**reusability:**
- modular, small components
- common patterns/templates
- low coupling / high cohesion
- closed architecture
- encapsulation
- generalization

**expandability**
- easily maintained
- well documented
- not over-designed (keep it simple stupid KISS)
- well designed database
- layered
- well organized
- flexible architecture
- standards

**portability**
- well designed boundary classes
- modular with device independent sections
- generalized architecture
- standards
- basic data structures
- encapsulated
- polymorphism esp. platform interfaces

The above is not a complete list of possible answers!

There were some answers that were too vague for marks:
- object-oriented
- good interfaces

Since there were quite a few answers possible, this question was marked by deducting wrong answers.
Section B  

Answer only three (3) of the following four questions. 
Answer the questions in the examination booklet.

THE ANSWERS TO SECTION B ARE IN A SEPARATE DOCUMENT.

B1.  

Draw an activity diagram that models the following scenario for a point of sale system. [15]

- the sales clerk enters item codes until all the customer’s purchases are recorded
- the subtotal, taxes and total amount due are calculated
- the customer can choose to pay with cash or a credit card
- if the customer chooses to pay by credit card, a credit check is done
- if the customer’s credit card is declined or the customer has insufficient cash, the sale is voided
- if the customer can pay, the payment is recorded and a receipt is issued to the customer

B2.  

Draw a use case diagram that models the following small system. [15]
Refine your use case to show at least one <<extend>> and at least one <<include>> relationship.

- John is an enthusiastic gardener who earns money by doing gardening work for his neighbours
- John buys the materials (fertilizer, mulch, etc.) that he needs, and charges his customers for both labour and materials
- some customers make appointments when they want work done and pay when John completes the work
- some customers set up a schedule (usually weekly) for regular work and pay when John sends them a monthly invoice for work done during the month

B3.  

Draw a state chart diagram for a bank account that has the following characteristics. [15]

- a new account cannot be used until a credit check has been done on the account holder
- the account holder can withdraw up to $500 per day
- an account is active as long as there is money in it and there is no “hold” on it
- a “hold” is put on an account if the balance becomes negative or the account has been inactive for 2 years
- an account with negative balance can be reactivated by a deposit that brings the balance to zero or above
- an account that has not been used for two years can be reactivated by approval from the bank manager
- an account that has been inactive for 6 years is closed

B4.  

Draw either a sequence diagram OR a collaboration diagram for the following use case. [15]
Do not try to handle exceptions (such as incorrect PIN numbers, overdrafts, etc.)
Include a boundary class and a control class in your diagram.

Use case: withdraw cash (from a personal chequing account) at an ATM (automatic teller machine)

<table>
<thead>
<tr>
<th>actor action</th>
<th>system response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. enter card into ATM</td>
<td>2. request PIN (personal identification number)</td>
</tr>
<tr>
<td>3. enter PIN</td>
<td>4. check for valid/active account</td>
</tr>
<tr>
<td>5. display options</td>
<td></td>
</tr>
<tr>
<td>6. select withdraw option</td>
<td>7. display accounts attached to the card</td>
</tr>
<tr>
<td>8. choose the account to withdraw from</td>
<td>9. ask for amount to be withdrawn</td>
</tr>
<tr>
<td>10. enter amount to be withdrawn</td>
<td>11. dispense money</td>
</tr>
<tr>
<td>12. ask if a printed receipt is wanted</td>
<td></td>
</tr>
<tr>
<td>13. enter decision</td>
<td>14. issue cash</td>
</tr>
<tr>
<td>15. return card to customer</td>
<td>16 remove card from ATM</td>
</tr>
<tr>
<td>17. print receipt (if requested)</td>
<td>18. update account balance</td>
</tr>
<tr>
<td>18. update account balance</td>
<td></td>
</tr>
</tbody>
</table>