

Answers to selected exercises

Multiple choice questions: Recommended marks are given in brackets after each possible answer. Note that some wrong answers may accrue negative marks.

Answers to most (but not all) questions without answers can be found in the text.

Chapter 1

1. Choose **two** characteristic features of OO

- a) Polymorphism (0)
- b) Inheritance (2)
- c) Reusability (-1)
- d) Abstraction (1)
- e) Encapsulation (2)
- f) Genericity (-1)
- g) Information hiding (0)
- h) Object identity (1)
- i) Dynamic binding (0)

2. Which artificial intelligence concept is closest to the idea of an object?

- 1. Slot (0)
- 2. Inference engine (-1)
- 3. Knowledge base (-1)
- 4. Frame (3)
- 5. Facet (0)
- 6. Rule (0)

3. What is the difference between the following? (*sample sentences given here*)

a) an instance and a class

Classes are collections of instances, which are individual things. A class is also a template for creating instances.

b) a data type and a class

Data types are predefined primitives; classes can be defined by the user.

c) a class and a rôle

Instances can adopt different rôles during their lifetime.

d) an object type and an entity type

Data types have no operations, unlike classes and object types.

e) a class and a component

In most programming languages outbound interfaces are not made explicit for classes, although in analysis there *should* be no difference.

f) dynamic binding and polymorphism

The former is one way of implementing the latter.

4 Requirements Engineering and Rapid Development

g) genericity and inheritance

Generic containers allow their content's type to vary, rather than their subclasses.

h) inheritance and classification

Classification refers to the relationship between instances and their classes/types, while generalization (or specialization) refers to the analogous relationship between classes; both are often referred to as inheritance. *Students may also discuss the difference between implementation and type inheritance but that was not the point in the chapter.*

4. Justify including attributes in class descriptions.

Attributes and associations are really exactly the same thing. Convenience is the main criterion. Don't clutter diagrams with primitive concepts. Ultimately, everything becomes a 'pointer'. Extra mark for discussing the bidirectionality issue.

6. Define object identity.

Refer to definition via index.

7. Define and give examples of a:

a) class attribute/method;

Average height/age of a class of people.

b) instance attribute/method.

Height/DoB of a person.

8. What is multiple inheritance? When should it be used?

The phenomenon of something acquiring characteristics from two parents in an inheritance structure. Good for modelling. Dangers should be pointed out; e.g. name conflicts. Discuss interface v. class multiple inheritance a la Java. Extra marks for spotting that rules can disambiguate.

Chapter 2

1. What is the essential difference between object-oriented and conventional systems?

1. OO is more reusable (0)

2. OO uses inheritance (0)

3. OO systems reduce development costs (-1)

4. Changes to data structures are localized (3)

5. OO is more fashionable (1)

2. Name up to eight benefits of object technology in general.

Localized maintenance, reuse, extensibility, conceptual integrity, seamlessness, etc.

Refer to chapter summary.

3. Name up to five pitfalls.

New things to learn, etc.

Refer to chapter summary.

Chapter 3

1. Which programming language is associated with the beginnings of OO?
 - a) Algol 68 (-1)
 - b) PL/1 (-1)
 - c) Simula (3)
 - d) Pascal (-1)
 - e) Ada (-1)
 - f) Lisp (0)

2. Which of the following did **not** help with the development of Smalltalk?
 - a) Adele Goldberg (0)
 - b) Alan Kay (0)
 - c) Bjarne Stroustrup (1)
 - d) Dan Ingalls (0)
 - e) Brad Cox (1)

4. Why did Java succeed so quickly, while Eiffel didn't?

Might mention C-like syntax and cleverer, better funded marketing.

5. What are the major benefits and pitfalls associated with C++?

Might mention performance and memory/resource leaks. Might also discuss the differences between languages for application and systems programming.

6. Distinguish between Java Beans and Enterprise Java Beans.

Might mention different levels of complexity and the various distribution issues.

Chapter 4

6. Describe the OMA.

This is the OMG Object Management Architecture that underpins CORBA.

9. What is the name given to the OMG standard for inter-object communication?
 - a) Object Services Architecture (-1)
 - b) Common Object Request Broker Architecture (3)
 - c) Object Interface Definition Standard (0)
 - d) None of the above (-1)

12. Why is having XML and middleware not enough to ensure successful Enterprise Application Integration? Discuss the problems to be expected on a typical EAI project.

This question is about the need for domain modelling to capture the semantics of inter-component communication.

6 Requirements Engineering and Rapid Development

14. What does the S in XSL stand for?
Stylesheet

15. Which concept is said to help with migration to object technology but safeguards investment in existing code?

- a) Coupling (-1)
- b) Object wrapper (3)
- c) Overloading (-1)
- d) Cohesion (-1)
- e) Polymorphism (-1)

Chapter 5

1. In object-oriented databases, what is the name given to the ability of objects to exist after an application program has terminated?

- a) Static Binding (-1)
- b) Persistence (3)
- c) Two-phase commit (-1)
- d) Object integrity (-1)

3. How can inheritance be implemented in a relational database?
Using the concept of entity-subtypes or as shared IDs with a secondary index.

7. Why can the performance of a pure object-oriented database exceed a relational or object-relational one?

Because of the need to perform many joins to reconstruct a complex object in a relational engine. OODBs traverse pointers instead of doing joins. *Include explanation of the architecture of all three options in this answer.*

Chapter 6

How many published methods or fragments of methods for OOA/D have been published?

- a) (-1)
- b) between 13 and 25 (0)
- c) (0)
- d) between 26 and 44 (0)
- e) (0)
- f) over 44 (3)

Which of the following OOA/D methods is associated with Jim Rumbaugh?

- a) CRC (-1)
- b) HOOD (-1)
- c) Ptech (-1)

- d) ObjectOry (-1)
- e) OMT (2)
- f) UML (1)

3. Define 'object' (one sentence). Name the four components of an object.
An object is anything with unique identity throughout its life.

Identifier

Attributes and associations (state)

Operations (methods, behaviour)

Rules (invariants, assertions)

4. What is the difference between types, classes, instances and rôles?

Types represent ideas and have no implementation (e.g. Person), classes represent collections of things with an implementation (People), instances are members of these collections (fredBloggs) and rôles are dynamic types taken on by instances (Holidaymaker). *Also mention the difference between types and interfaces (invariants) in a good answer.*

5. What is a 'facet'? Give three completely different examples.

An attribute of an object feature.

Keyword of an object description (ID)

Range constraint or default (init) value for an association.

Pre-condition of an operation

Régime of a ruleset

6. Why is analysis more important for OO systems?

Because of the need for stable interfaces.

9. Name the four principal structures of an object model.

Association, Inheritance (generalization, etc.) , Aggregation (composition) and Usage (uses, message passing).

10. Redraw the inheritance structure of Figure 6.16 with due attention to different discriminators.

Hint: ProductType. PaymentMethod

11. What is the difference between a wrapper and a subsystem or package in most OOA methods?

Wrappers enforce encapsulation.

26. Why is analysis important for OO programmers?

See Q 6.

27. Does God throw exceptions?

No, but she might do if she existed.

Chapter 7

4. Distinguish between a patterns catalogue and a pattern language, giving examples. A pattern language lets you string patterns together to create design 'sentences' that help with the construction of a solution.

6. Write patterns to deal with the following situations:

a) the need to inform all users of e-mail of a change to their e-mail addresses, when some may be on holiday;

b) users in a workshops continually disagree with each other;

Hint: look up 'lead user' in the index.

c) management resist object technology because they consider it too risky.

10. Mini project

Produce a framework template for school timetable preparation: allocating suitable classrooms, qualified teachers, ten subjects and five one-hour time-slots per weekday. Include all invariants; e.g. Chemistry needs a classroom with sinks and there must be at least five hours of Maths and English per week. Are there any rules that are hard to express in OCL? Why is this? Apply the same framework to factory production scheduling.

This question is advanced and requires some knowledge of AI.

Chapter 8

5. What is the first thing a facilitator does in a workshop? What is the last?

First: Say good morning, introduce participants, agree the mission.

Last: Get sign-off

6. Name ten of the key tasks of a workshop – in order.

Hint: see checklists on pages 447-450.

7. Discuss the use of voting in workshops.

Bring out the need for speed and consensus as opposed to more scientific methods such as pairwise ranking.

10. What is the grammatical structure of an atomic task? Define atomic. How else can scripts be specified?

a) SVDPI

b) Its atomic when it is a single (possibly long) sentence that if decomposed further would need to introduce terms outside the domain ontology (vocabulary).

c) By numbered lists of natural language statements or as sequence diagrams.

11. Discuss the validation of requirements models.

a) How is the Business Object Model tested?

Walkthrough against the use cases in the TOM

b) How are objectives tested?

By measures

c) How is the Task Object Model tested?

By cross-reference to the business objectives and by the walkthrough.

12. How is a Task Object Model transformed into a Business Object Model?

By creative thinking, reference to domain knowledge and textual analysis.

How is a Business Object Model transformed into an Implementation Object Model?

See *Chapter 6 and (esp.) 7*.

14. Complete the following well-known phrases and sayings:-

Better Systems faster!

I keep six honest serving men,

They taught me all I knew.

Their names are What and Why and When

And How and Where and Who.

18. Mini-project

A dealer has already found an opportunity to deal in the FX market with a certain counterparty. The dealer is obliged to execute the deal only within given counterparty limits and within her personal position limits. They must complete the deal once they have agreed it verbally. If these conditions are satisfied then:

The dealer strikes a bargain with the counterparty at agreed terms and rates;

A confirmation must be sent to the counterparty;

The Settlements Department must be notified of the deal so that they can arrange payment. Settlements also provide standing settlement instructions and notify the Accounts Department of the deal;

Accounts post on the deal date and on its value date;

The dealer's position and the counterparty's limit availability must be adjusted to reflect the new position.

Based on the above scenario, draw a business process model for the above situation, including agents and conversations and the tasks associated with them. Write a task script for one of the tasks. Sketch a Business Object Model using any elements that seem appropriate. Try to find the key objects and their four structural relationships: classification, usage, composition and association. Make up some class cards. If you are part of a group try to walk through and debug your model. Use the walkthrough to produce sequence charts and/or activity diagrams.

This isn't as hard as it looks. You should end up with about 8 objects including Deal, Counterparty, CptyAvailability (=CptyPosition – CptyLimit), etc. Note the red herring Accounts stuff: value date etc. The main point is to get to a walkthrough.

Chapter 9

10 Requirements Engineering and Rapid Development

1. (a) Why would a company wish to standardize on a development process?

People always follow some process (JFDI)

- ad hoc and informal
- individual
- no guarantees of same result

Standardization

- follow the same steps
- group understanding and skills
- common tools and vocabulary

Predictable

- different teams can come up with similar results (staff substitutable) (?)
- less 'work till you drop' to finish the project (death march)

Repeatable successes

- can be followed for different projects

Process improvement

Basis for measurement and estimation

Quality

Seamlessness (OO wall to wall)

Traceability

Lower time to market (?)

A high level of quality is ensured within the system as all artifacts produced during the project lifecycle are reviewable thereby allowing for quality checks.

Bad design is reduced and/or eliminated since the system cannot be built without an architecture specification and design.

Project risk is reduced since expectations are clearly set as to what artifacts are produced at what stage, therefore if something is not delivered it prompts remedial action.

Sets expectations; each person playing a role during the project lifecycle is clearly aware of her/his duties and expected activities/outputs.

Allows reuse of designs/documentation thereby reducing the time to market in subsequent projects.

Promotes a sharing of knowledge among those that participate in the project thereby raising the quality and abilities of those participants. Professionalism and technical competence increases across the board given the same base levels of abilities among employees were process not present.

Improved morale since the chaos and stress associated with failing/late projects is reduced to a minimum. If process is executed effectively the attrition rates within a company are dramatically reduced with an increase in pride of workmanship.

b) What disadvantages might they see?

Lack of creativity

Red tape

... leading to Sabotage

etc.

2. Enumerate the benefits of time-boxes.
 Imposes management control over ripple effects and uncontrolled iteration
 A usable system as both end-point and deliverable
 No distinction between production, evolution and maintenance
 Reduced time to market
 Prevents paralysis by analysis, errors due to delay, spurious requirements and implementation shock
 Motivates teams
3. Give examples of product and process metrics. How might they be related?
 Product: SLOCs, function points, task points, mean time to failure, bugs reported per annum, etc.
 Process: time to develop, PhDs per developer, average overrun on projects, etc.
4. Define the term ESTIMATION MODEL. Give an example of one.
 An estimation model is a mathematical relationship between process and product metrics; e.g. it cost \$60 to produce one line of COBOL.
10. How long does it take to go from the end of a requirements workshop to implementation?
 It depends on many factors (*discuss them*). Anything from 11 hours to six months absolute maximum.
14. Complete the phrase: Users should be available but not too available.
16. Devise a menu similar to that of Figure 9.22 that supports priming without using Greek characters.
 One way might be to duplicate the names: once in plain text and once in the font.
Extra marks for finding another solution.
18. Should a mouse move the pointer further when the user's hand moves faster? If not, why not?
 To do so would violate the Idempotency principle and makes switching between operating systems very hard indeed.