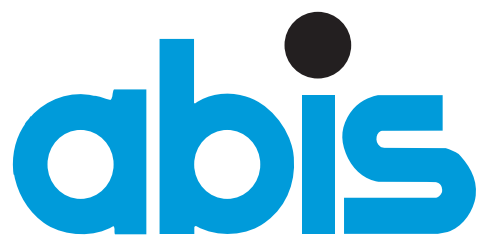


# Self-test OO concepts and techniques

Document: e0611test.fm

3 September 2019

ABIS Training & Consulting  
Diestsevest 32 / 4b  
B-3000 Leuven  
Belgium



TRAINING & CONSULTING

# INTRODUCTION

In order to get a reasonable indication for the prerequisites of the Java programming and OO analysis and design with UML courses, the following questionnaire might help to check your personal knowledge of the various topics discussed in the OO concepts and techniques course.

In this test, 20 questions are asked. For most questions, *there is only one answer possible*. If multiple answers are possible, it will be clearly indicated. The question is answered correctly, if and only if all correct answers are given.

There are no real trick questions (at least not intentional) but read question and answer attentively.

You must take into account that this test will take you about 10 minutes.

You can find the right answers and guidelines for the evaluation at the end of this document.

# QUESTIONS

1. Which list of characteristics describes best the **Object Oriented** programming?

- (a) layered architecture, concurrency, high coupling
- (b) encapsulation, inheritance, polymorphism
- (c) data modelling, subtypes, services
- (d) classes, objects, high performance

2. What is true about **objects**? [2 answers]

- [a] An object is initialized by 'overriding'
- [b] An object is an instance of a class.
- [c] An object can be accessed in an application via the object reference.
- [d] An object defines the behaviour of its class.

3. How do you **extend a class**?

- (a) by allocating more memory to the class
- (b) by making a more specific class
- (c) by creating extra instances of the class
- (d) by creating a superclass

4. What does the following UML diagram show?



- (a) a composition
- (b) an association
- (c) an inheritance
- (d) a compound interaction

5. Which statement about **inheritance** reflects best the OO philosophy?
- (a) Inheritance shows how a class changes over time.
  - (b) Inheritance shows how messages are passed between classes.
  - (c) Inheritance is a programming technique to reduce the amount of code.
  - (d) Inheritance allows similarities and dissimilarities to be modelled clearly.
6. What is true about **classes** and **objects**?
- (a) Every class is instantiated in one or more objects.
  - (b) Objects change class over time due to polymorphism.
  - (c) A class supplies the program code for objects to be executed at run-time.
  - (d) Classes are shown on conceptual diagrams and then the objects are written by programmers.
7. What is true about **abstract** and **concrete classes**?
- (a) An abstract class is inherited.
  - (b) A concrete class has a concrete use case.
  - (c) An abstract class is never instantiated
  - (d) Abstract classes are not shown on a class diagram.
8. In terms of object oriented programming, the use of **polymorphism** means
- (a) that a client class does not need to be aware of the particular subclass that actually implements the method used.
  - (b) that objects change class over time.
  - (c) that a single object is morphed into a number of database tables.
  - (d) that a subclass can override the methods of a parent class only if is not abstract.
9. The **hierarchy** of subclasses and superclasses allows for *[2 answers]*
- [a] making associations between the different classes.
  - [b] implementing polymorphism
  - [c] operating safely on collections of objects from different classes in the hierarchy.
  - [d] changing object behaviour dynamically, by pointing to any other class in the hierarchy.

10. What is true about **UML**?
- (a) UML is a standardised approach for use case modelling.
  - (b) UML supplies a set of notations used in the design of applications.
  - (c) UML is a methodology for designing and maintaining computer systems.
  - (d) UML is an alternative for RUP and Agile.
11. Which technique do most Object Oriented methods propose to describe the **functionality and scope** of a system?
- (a) use cases
  - (b) a statechart diagram
  - (c) flow charts
  - (d) Entity Relationship Diagrams
12. What does **encapsulation** mean In terms of object orientation?
- (a) The code is layered; typical layers are: user interface, business logic and persistence.
  - (b) Logically related data and procedures are grouped in a class, while hiding the internal behaviour and data structure for other objects.
  - (c) Objects can only access their own attributes.
  - (d) Encapsulation establishes the implementation, after that only the public interface can be adapted.
13. What does **High Cohesion** mean?
- (a) It means that all the classes work very closely together, which is a good principle.
  - (b) It means that all the classes work very closely together, which should be avoided.
  - (c) It means that each class does one thing well, which is a good principle.
  - (d) It means that each class does one thing well, which should be avoided.
14. What is true about **Design Patterns**? *[2 answers]*
- [a] Design Patterns define the graphical layout of a web application.
  - [b] Design Patterns were invented specifically for Java programming.
  - [c] Design Patterns were made popular by the Gang of Four (GoF).
  - [d] Using Design Patterns improves the maintainability of an application.

15. What is true about **Testing**? [2 answers]
- [a] A good OO application can only be tested if all components are built.
  - [b] Testing is an integral part of OO methods such as RUP and Extreme Programming.
  - [c] Black box testing of a component requires knowledge of its concrete implementation.
  - [d] Use cases are used to set up test scenarios.
16. What is true about the **RUP**?
- (a) According to the RUP, implementation should start when design is ready.
  - (b) The RUP is an iterative and incremental approach.
  - (c) According to the RUP, design can only start when implementation is ready.
  - (d) The RUP discourages the use of fixed deadlines (time boxing), because quality control and testing may not be complete then.
17. JDBC, Hibernate, ADO .NET and ADO .NET Entities are standards for
- (a) design objects
  - (b) remote network components
  - (c) persistence
  - (d) database administration
18. In OO languages the decision about which code is to be executed, is often only made at run-time. How is this principle called?
- (a) static binding
  - (b) dynamic binding
  - (c) dynamic typing
  - (d) strong typing
19. What is the purpose of **O/R mapping**?
- (a) Establishing the link between the object representation and the storage in a relational database.
  - (b) Mapping the relationships between the application objects.
  - (c) Organizing the OO application objects in a hierarchical folder structure.
  - (d) Giving a high-level description of the application objects and their functionalities.

20. What is true about **frameworks**? [2 answers]

- [a] Frameworks are based on distributed processing.
- [b] A framework is a standardized development environment.
- [c] A framework provides a reliable context and components for building applications.
- [d] Frameworks enable consistency between projects.

# EVALUATION.

Here are the correct answers to all questions:

1. b
2. b c
3. b
4. b
5. d
6. c
7. c
8. a
9. b c
10. b
11. a
12. b
13. c
14. c d
15. b d
16. b
17. c
18. b
19. a
20. c d

Give yourself 1 point for each correctly answered question (for multiple answer questions, this means that **all** correct answers must be marked).

If you answered less than 50% of the questions correctly, please follow the [OO concepts and techniques](#) course, before you start the Java or UML courses.

If you answered between 50% and 75% of the questions correctly, you know already some concepts, but you will still learn a lot in the [OO concepts and techniques](#) course.

If you answered more than 75% correctly, then the [OO concepts and techniques](#) course is probably not useful for you any more. You are ready to start the Java curriculum ([Java concepts](#), [Java programming](#), ...) or the [OO analysis and design with UML](#) course.