



Exam Questions

Object-Oriented Design, IV1350

Maximum exam score is 100, grade limits are as follows.

Score	Grade
90	A
80	B
70	C
60	D
50	E

The exam questions will be a subset of the questions below. The exam may contain one or two (but not more) questions that are not listed in this document.

Object-Oriented Analysis

1. Explain in detail the goal of analysis. (5p)
2. Explain in detail the purpose of a system sequence diagram? (5p)
3. Create a system sequence diagram for the following scenario. *Scenario will be presented at exam.* (5p)
4. Explain in detail what a domain model is and what function it fulfills. (5p)
5. Thoroughly describe two methods that can be used to find class candidates for a domain model. (5p)
6. Specify how you can determine which associations to include in a domain model. Give a concrete example of an association you would like to include in a domain model, explain why you would like to include it. Also give a concrete example of an association you would *not* like to include in a domain model, explain why you would not like to include it. (5p)
7. Specify a guideline for when an abstraction should be modeled as a class and when it should be modeled as an attribute in the domain model. (5p)



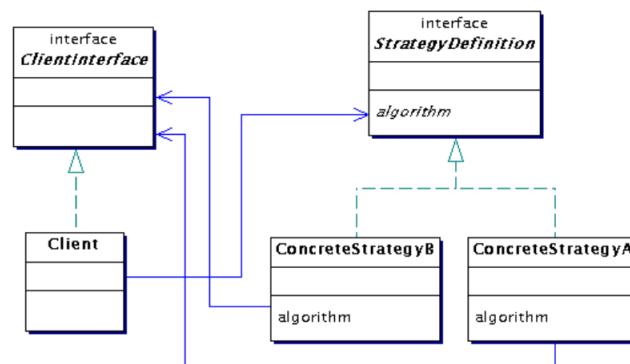
8. For each of the abstractions in bold below, motivate if it should be represented as a class or as an attribute in the domain model. *Abstractions will be presented at exam.* (5p)

Object-Oriented Design

9. Explain in detail the goal of design. (5p)
10. Specify when each of the four different accessibilities (public, private, etc) should be used. Give a concrete example illustrating usage of each accessibility. It is sufficient to explain the examples in words, you do not have to write code (but you may if you wish). (5p)
11. Thoroughly explain the difference between public interface and implementation. (5p)
12. Which parts of the code are *public interface* and which parts are *implementation* in a layer consisting of many packages, classes and interfaces? You shall thoroughly state exactly which classes, methods, etc belong to the public interface, and which belong to the implementation. (5p)
13. The following design has low cohesion. Explain why and suggest a change that will give high cohesion. Also motivate why your change gives high cohesion. *Design will be presented at exam.* (5p)
14. Explain the meaning of encapsulation and give an example of the benefits of encapsulation. How can you decide if a program makes appropriate use of encapsulation? (5p)
15. Explain the meaning of cohesion and give an example of the benefits of high cohesion. How can you decide if a program has high cohesion? (5p)
16. Explain the meaning of coupling and give an example of the benefits of low coupling. How can you decide if a program has low coupling? (5p)
17. Explain why it is not good that a method has greater dependency on another class than the one in which it is found. Also explain how to solve this problem. (5p)
18. Thoroughly explain the meaning of polymorphism. Which are the advantages of using polymorphism? (5p)
19. Suppose we have a class, call it **Server**, which has some desired functionality. Suppose further that we want to be able to replace it with another class, **OtherServer**, which performs the same task but in a different way. It must be possible to switch between **Server** and **OtherServer** without having to recompile the classes that call them. Write a program which solves this problem with the help of polymorphism. (10p)



20. Give an example of how you can use an interface to give different classes a common *public interface*. Write a program that demonstrates how objects of the classes can be called using polymorphism. Your program shall include the interface, the classes implementing the interface and the class that calls the implementations. Nothing more is required. (10p)
21. The classes **Person** and **Airplane** shall be able to print their current state on a printer. Show how to design this using an interface for the *printout* functionality. Draw class and interaction diagrams, which include the class initiating the printout, and explain the diagrams. Explain the advantage of this solution over a solution without the interface. You do not need to use correct class or method names to manage printouts. (5p)
22. a) Suppose we need to store instances of the classes **Car** and **BankAccount** in a database. To obtain low coupling it may seem as if the functionality to save an instance should be in precisely those classes, since they have the information that shall be saved. Why is it unwise to place the database calls there? (5p)
- b) Describe a better way to manage the database calls. (5p)
23. Explain what a design pattern is. (5p)
24. Give a *concrete* example illustrating the pattern (*pattern will be specified at exam*). Write a program illustrating the example, you are only required to create classes relevant for the pattern. Explain usage of the pattern in your example. (10p)
25. Suppose we use the Strategy pattern and that classes **ConcreteStrategyA** and **ConcreteStrategyB** need to call the client to perform the algorithm they implement. This can be solved as illustrated in the diagram below. What is the reason to write the interface **ClientInterface**? (5p)



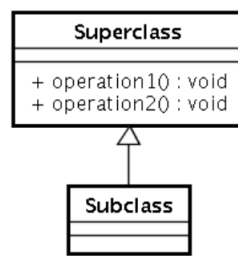
26. Suppose we have a Factory that can create objects of various classes. It has methods like:



```
public ProductA createProductA()  
public ProductB createProductB()  
...
```

One problem with this is that every time we want to enable the Factory to create objects of a new class, we have to add a new method. Describe another way to write methods in a Factory, which removes the need to add a new method when the Factory shall be able to create objects of a new class. Explain the pros and cons of the two ways to write a Factory. (5p)

27. What is the reason to use a Singleton instead of a class with only static members and never instantiate any object of that class? (5p)
28. Is there any occasion when it is better to make everything in a class static instead of making it a singleton? If yes, which? (5p)
29. Name three important characteristics of a controller. The question concerns characteristics specific to a controller, it is not correct to answer *high cohesion, low coupling and good encapsulation*. (5p)
30. Why can it cause problems to inherit classes that are not intended to be inherited? Give two different reasons. (5p)
31. What conditions must be met to make it appropriate to create a new class hierarchy using inheritance? Neither superclass(es) nor subclass(es) exist before, the question is about designing a new class hierarchy. (5p)
32. Suppose you are about to code a particular class and find that you have written another class with nearly the functionality that you now are developing. This means that the two classes in large parts contain the same code, but in smaller parts have different code. Explain in detail why this may be a problem and how to solve it. (5p)
33. Suppose you are about to code a certain functionality, and that in a program library you are using there is a class that has almost the sought functionality. Is there any reason to still use that class and, if so, how can you add the functionality it is missing?
34.
 - a) Suppose you have the class hierarchy shown below, how can you change the program to use composition instead of inheritance? The functionality should not be changed. Answer with code using composition. (5p)
 - b) Explain why composition is more appropriate than inheritance if the superclass is not intended to be inherited. (5p)
35. Explain why exceptions should only be used when an error has occurred. (5p)



36. Thoroughly explain why errors should always be reported using exceptions, and never using return values. (5p)
37. Suppose you have a method (*method and its functionality will be specified at exam*). Which results shall be return values and which shall be exceptions? (5p)
38. Suppose a method that searches in a customer register shall be written. The method shall return found customers' names and phone numbers. One idea is to have the method return an array of strings where the first element is the first customer's name, the second element is the first customer's phone number, the third is the second customer's name, the fourth is the second customer's phone number, followed the same way by remaining customers' names and phone numbers. Why is this a bad solution? Suggest a better solution! (5p)
39. Thoroughly explain what should be tested by unit tests. (5p)
40. Explain in detail which unit tests should be written for the following code (*Code will be given at exam*). You do not have to program unit tests (but you may if you wish). (5p)

Object-Oriented Programming

41. Implement the following class diagram/communication diagram/sequence diagram in correct code. You are required to use the strongest possible encapsulation. (*diagram will be presented at exam*). (10p)
42. Explain why duplicated code may cause problems (it's not enough to write that it "is almost impossible to maintain") and how you can eliminate the duplicated code. (5p)
43. Explain why long methods may cause problems and how you can split long methods. (5p)
44. Why is it inappropriate that methods have long parameter lists? Explain two ways to shorten long parameter lists. (5p)
45. Suppose your program consists of a few large classes that have many fields with primitive data. Why is this a problem and what can you do about it? (5p)



46. Why is it a problem if a large part of the input parameters to methods are primitive data instead of objects. Explain how you can solve that problem. (5p)
47. Describe where in the code you would write comments. Explain both why you want comments where you place them and why you do not want comments in other places. (5p)
48. Assume that your program contains a user interface where the user enters an ID as a string. Specify why it is a good practice to convert the string to an object that represents the ID. Which object should be responsible for the conversion from string to ID? Motivate! (5p)
49. Explain how you want to name packages and how you want to divide a program in packages. Suppose you're talking about a new program, you do not need to worry about any previous package structure. (5p)
50. How would you like to package utility classes? Indicate in which package(s) you would place a class that parses strings and a class that handles URLs. (5p)

Object-Oriented Architecture

51. Explain in detail how you can use the patterns MVC and Layer to create a good architecture. (10p)
52. Explain in detail which the three different parts of the MVC pattern are, what they should contain and how they can communicate with each other. (5p)
53. Explain in detail why objects in the model never shall call objects in the view when the MVC pattern is used. (5p)
54. Explain in detail why objects in the model never shall contain code for receiving input directly from the user when the MVC pattern is used. (5p)
55. Describe the integration layer. Explain what functionality should be in the layer and why it is good to introduce the layer. Motivate your answer by indicating which of the properties encapsulation, high cohesion, low coupling, polymorphism and inheritance that are enhanced by the integration layer, also explain how the properties are enhanced. (5p)