The maximum exam score is 100, grade limits are as follows.

<table>
<thead>
<tr>
<th>Score</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>A</td>
</tr>
<tr>
<td>80</td>
<td>B</td>
</tr>
<tr>
<td>70</td>
<td>C</td>
</tr>
<tr>
<td>60</td>
<td>D</td>
</tr>
<tr>
<td>50</td>
<td>E</td>
</tr>
</tbody>
</table>

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 the goal of analysis. (5p)

2. What is 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 what a domain model is and what function it fulfills. (5p)

5. Describe two methods to find class candidates to a domain model. (5p)

6. Specify how you can determine which associations to include in a domain model. (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. Motivate why the following abstractions should be represented as a class or why they should be represented as an attribute in the domain model. *Abstractions will be presented at exam.* (5p)
Object-Oriented Design

9. Explain the goal of design. (5p)

10. Specify when each of the four different accessibilities (public, private, etc) should be used. (5p)

11. Explain the difference between public interface and implementation. (5p)

12. Which part is the public interface and which part is the implementation in a subsystem consisting of many classes and interfaces? (5p)

13. The following design has low cohesion. Explain why and suggest a change that will give high cohesion. The design will be presented at exam. (5p)

14. Explain the meaning of encapsulation and give an example of the benefits of encapsulation. (5p)

15. Explain the meaning of cohesion and give an example of the benefits of high cohesion. (5p)

16. Explain the meaning of coupling and give an example of the benefits of 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. (5p)

18. Explain the meaning of polymorphism. (5p)

19. What are the advantages of using polymorphism? (5p)

20. Give an example where polymorphism is used. Implement the example in a program, you do only have do code classes relevant for the polymorphism. (10p)

21. 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)

22. 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. (10p)

23. The classes Person and Airplane shall have the property to to print information about themselves on a printer. Show how to design this using an interface for the property to print. Draw class and interaction diagrams. You do not need to use correct class or method name to manage printouts. (5p)
24. 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)

25. Explain what a design pattern is. (5p)

26. Give a concrete example illustrating the pattern (pattern will be specified at the exam). Write a program illustrating the example, you are only required to create classes relevant for the pattern. (10p)

27. 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?

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

   ```java
   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)

29. 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)
30. 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)

31. Name three important characteristics of a controller. (5p)

32. Why can it cause problems to inherit classes that are not intended to be inherited? Give two different reasons. (5p)

33. What conditions must be met to make it appropriate to create a new class hierarchy using inheritance? Neither superclass nor subclass exist before, the question is about designing a new class hierarchy.

34. 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)

35. 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?

36. 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 class diagram or code. (5p)

```
Superclass
  + operation1: void
  + operation2: void
    
Subclass
```

b) Explain why composition is more appropriate than inheritance if the superclass is not intended to be inherited. (5p)

37. Explain why exceptions should only be used when an error has occurred. (5p)

38. Explain when an inner class is appropriate. Give a concrete example of using an inner class. (5p)
Object-Oriented Programming

39. Implement the following class diagram/communication diagram/sequence diagram in correct code. You are required to use the strongest possible encapsulation. The diagram will be presented at the exam. (10p)

40. 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)

41. Explain why long methods may cause problems and how you can split long methods. (5p)

42. Why is it inappropriate that methods have long parameter lists? Explain how long parameter lists can be shortened. (5p)

43. 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)

44. 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)

45. 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)

46. 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)

47. Explain how you want to name packages and how you want to divide the source code in packages. Suppose you’re talking about a new program, you do not need to worry about any previous package structure. (5p)

48. 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

49. Explain how you can use the patterns MVC and Layer to create a good architecture. (10p)

50. Explain which the three different parts of the MVC pattern are, what they should contain and how they can communicate with each other. (10p)

51. Explain why objects in the model never shall call objects in the view when the MVC pattern is used. (5p)
52. Explain why objects in the model never shall contain code for receiving input directly from the user when the MVC pattern is used. (5p)

53. 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)