Expectation

- Object-Oriented (OO) software development
  - Program design, implementation and testing
  - OO concept
    - classes
    - objects
    - encapsulation
    - inheritance
    - polymorphism
  - The Java programming language
  - Problem solving
    - data structures
    - algorithms

Prepare for Exam

- Sample exam and solution in the LMS, and you can also find a past exam 2016s1 (from university's digital library)

- Format of the exam of this semester (2019s1)
  - It contains the same types of questions like the above sample and 2016s1 exams, plus the following new type. Also the distribution of marks may be different; questions will be different.

  - New type of questions: *Fill in the blanks*
    - Example: What is the name of the class that is the ancestor of all the classes in Java? ________

- Preparation: lectures, textbook, workshops, projects, sample exam, lots of practices of programming

Review

- 1: Introduction
  - What a java program looks like? How it works? How to compile and run it, etc
  - Basic operations: primitive types, identifiers, assignment statement, arithmetics, string, etc
  - Importance of software engineering (SE). The waterfall model.

- 2: Console Input and Output
  - System.out.println (printf, print), various formats
  - Input using the scanner class, nextInt (nextFloat…), nextLine, etc

- 3: Flow of Control
  - Boolean expressions: logical values, !, &&, ||, >, <, ==, precedence and association rules
  - Branching: if-else; multiway if-else; switch; break; continue
  - Loops: for, while, do-while; nested loop, infinite loop, debugging a loop

- 4: Classes I
  - Type, members (instance variables, methods), local/global variables, this, access permission (public, private), **overloading** (same name different signature), constructors
  - Modularity, information hiding, encapsulation

- 5: Classes II
  - Static methods and variables
  - References, privacy leak, mutable and immutable classes, equals, toString, packages
  - UML

Review

- 6: Arrays
  - Basic operations, references, string array, **No** multidimensional array.
- 7: Inheritance
  - Base/derived classes, **overriding**, super constructor
  - More access permission
- 8: Polymorphism and Abstract classes
  - Late binding (toString): except static, final, private methods; downcasting/upcasting; **No** clone methods
  - Abstract class: a class containing an abstract method; cannot define an object of an abstract class
- 9: Exceptions
  - Try-throw-catch; Exception class; getMessage(); checked/unchecked exceptions
- 13: Interfaces
  - Generalized selection sort example; Comparable interface

Review

- 10: File I/O
  - Streams, text file and binary file
  - Textfile: opening, reading (PrintWriter), writing (Scanner, BufferedReader)
  - File class (getName, setReadonly, delete, etc)
  - Binary files: ObjectInputStream/ObjectOutputStream, Serializable interface
  - **No** random access to binary files

- 14: Generics & ArrayList
  - ArrayList: basic operations, methods (add, set, get, etc), for-each loop
  - Generics: parameterized classes/methods
  - **No** coding required for generic methods

- Project

No test on: clone (copy is still examinable), inner class, random access to binary files
This is just a list of the important points. See the subject website for the coverage of the exam.