Relevant arguments for the exam of Functional Languages 20/21

--- the importance of modularization of programs and the ways in which functional programming enhances it. Examples are welcome. John Hughes paper.

----Haskell constructs for defining functions (where, let, case, guards, list-comprehension, etc.).

----Currification, higher order functions,

---- Chapter 7 with its exercises,

----foldr and foldl being able to use them for defining functions,

----notions about IO, why IO is not pure and how Haskell faces the problem,

----type classes and types and their relation

----type constructors and data constructors, the notion of kind

---- Chapter 12 with its exercises. Be ready to explain what Applicatives can do more than Functors and Monads more than Applicatives and to support what you state with examples.

---- the type State Transition,

---- exercises examined during the course, like the Hanoi towers, the acrobat, the relabelling of trees, balance of parentheses, etc.

---- Chapter 15 with its exercises 1-5.

---- Chapter 16 with its Exercises, excluding 9 and 10.

---- type inference