Some exercises forms
(non exhaustive list)

- Given a judgment $\Gamma \vdash t : \tau$ with "holes", fill the holes
  e.g. given Γ, τ, find t
  " Γ, t, " τ

  ex:
  $x : ? \vdash x\,(?) : \forall \alpha . ?$

- Construct a typing derivation for $\Gamma \vdash t : \tau$
  (for the λ-calculi where we have seen the rules)

- find the β / η / $\beta\eta$ normal form of some term t

- prove that $\nexists t . \vdash t : \tau$
  e.g. because τ leads to $\bot$ (& consistency)
  " τ leads to LEM

- theory questions:
  provide definitions / statements / proofs
  (those in the notes, only)

- prove a property on t, provided $t : \forall \alpha . \tau(\alpha) \to \sigma(\alpha)$, and parametricity
- state the "free theorem" (in its naturality form) for some polymorphic τ

- prove $\tau \simeq \sigma$ for some types
  - possibly, this can involve Yoneda

- prove that there is only one /two/ -- values for a type τ, under certain assumptions about the model

- verify some basic category theoretic notion: eg.
  - check F is a functor
  - check η is natural
  - check $\mathcal{C}$ is a category
  - ...
- prove some objects to be isomorphic $A \simeq B$
- prove two morphisms equal

- Given $F : |\mathcal{C}| \to |\mathcal{D}|$ defined on objects only, extend the definition to a functor $\mathcal{C} \to \mathcal{D}$
  Eg. $FX = X \times \mathbb{N} \times X$

- prove X initial

Note: some exercises may require basic properties of groups / vector spaces / topological spaces / sets

- state the induction principle on some recursive type
- use "cata g", at least informally, on some type e.g. use it to define some program computing some given function

Note: it is NOT required to know the syntax of Haskell / Coq / Java / Scala / ...

It is required only to have a good intuition about the ideas showcased in the examples we saw

- give the type for the eliminator of a (recursive) type
- ... and the type for the dependent eliminator

- given a Coq dependent match, describe intuitively why it type checks

- encode τ in a type system
  which allows (e.g.) only $\forall \alpha, \to$

Eg. $\tau = \mu \alpha . F \alpha$

$\rightsquigarrow \tau \simeq \forall \alpha . (F \alpha \to \alpha) \to \alpha$

$\simeq \ldots$