Qs for Monday

how type erasure works and why it is not a problem with wild cards? For the exam, is it enough to know the consequences in terms of generics eg the List<Number, List<Integer> issue. Or do we need to know the inner workings of erasure so we know why?

Is the common parent of List<Integer> and List<Number> List<?> or Object? What is the difference between List<?> and List<Object>? What does it mean to add NULL to List<?> why can you do it? What is actually happening? In general, need to understand how we can work with raw types.

Exceptions – should we only throw and catch checked exceptions, and then just catch runtime exceptions? Why do we need to explicitly throw run time exceptions if the machine will throw them anyway if necc? Why cant we just write code that catches them, without explicitly throwing them. What are the underlying mechanics that make us have to throw them?