In M3 (which you should work on first), you have written three styles of query: plain, "functional", and "typed".

Imagine that we need to extend our format to a new kind of elements called "disjunction" to take into account a new type of class expressions, namely disjunctions. Of course, this extension in our format may require an extension of our schema.

In 200-300 words, explain the advantages and disadvantages of updating your "get all class expression" queries to take into account this new element. In particular, explain which style of query is the "most robust" in the face of such format changes.

(As usual, if you are unsure whether you understand the exact meaning of a term, e.g., 'robust', you should look it up.)

There are three cases separately. A straightforward element path is described by a plain query. However, the robustness of plain query depends on xml file size. If xml file structure is relevantly simple, plain query might be good choice. It is hard to use plain query for a complicated xml file. In other word, plain query is not reusable unless xml file is simple.

On the contrary, “functional” query provide not only correct result but also reusable feature on xml. It provides strong readable feature as well. (i.e. “flwor” expression) Besides, it is a strong functional language. In this case, “functional” query can support a correct result with some modifies on query. Still, it depends on xml file. It will be very complicated if xml structure is complex. It is therefore not the most robust among three ways in the face of such format changes.

“Typed” query is the most robust. It provides more flexible and controls than the others. Note that this only affects modification on xml schema and doesn't change anything else on Xquery. Nevertheless, the schema will be very complicated if a new element added in xml. But it is the most robust on query level.