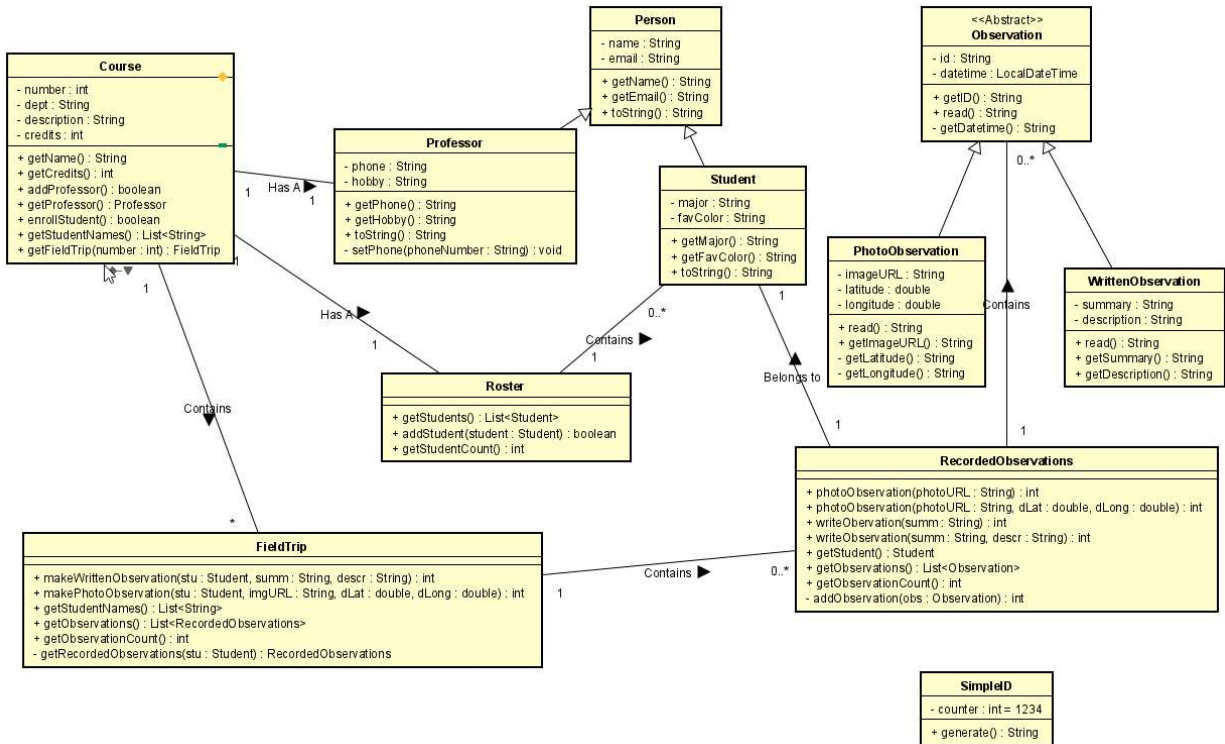
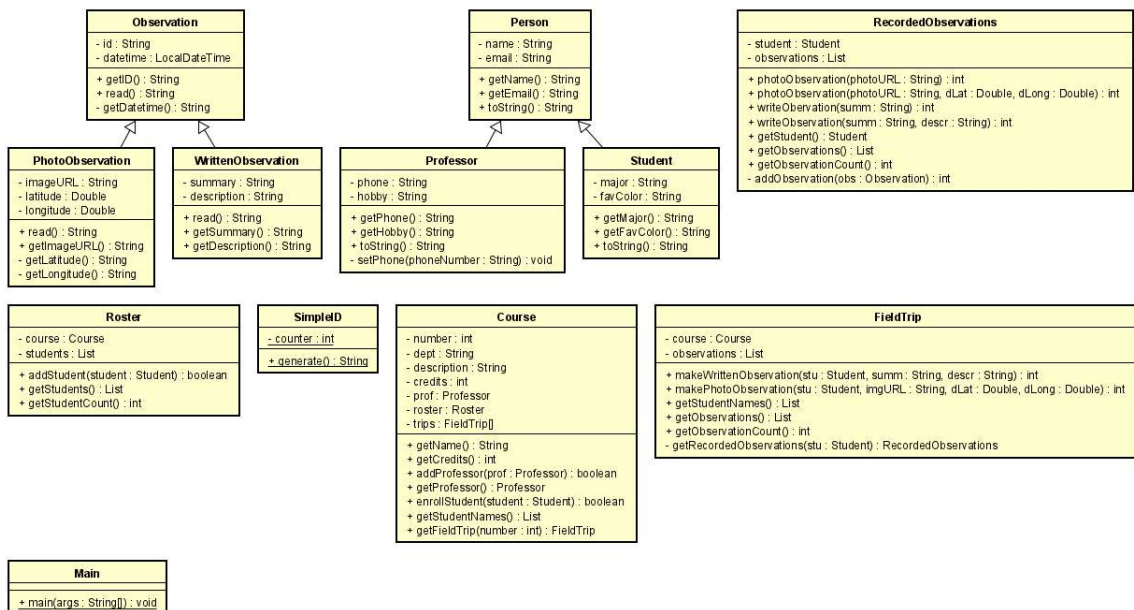


## Task 1: Manual Reverse Engineering

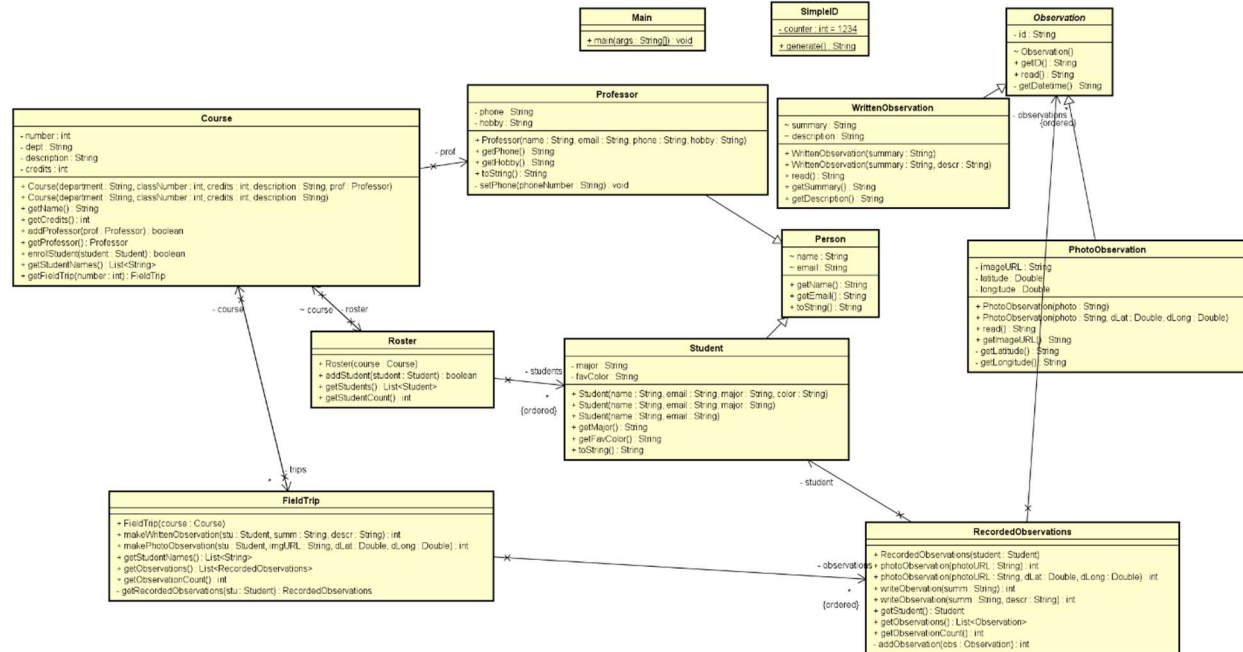


## Task 2: Automatic Reverse Engineering

### 2.1:



## 2.2



All three diagrams have a different level of abstraction. Explain the main differences.

The diagram from 2.1 has other classes listed as attributes. While the other two are using associations. 2.1 also does not make use of List<String>, instead it just has List. 2.2 is making use of both bi-directional and uni-directional associations. In my diagram I just used bi-directional associations.

Why do all three diagrams still represent the same system even though they look so different?

I feel like the biggest difference is that 2.1 does not use associations. My diagram and 2.2 use associations. 2.2 is making use of both types of associations.

Why is an association basically the same as having the attribute in the class?

I don't think they are the same. I feel like using the association means you have a using situation. Where you can use the other class inside of it. With associations you can have the 1 to 1 or 1 to many types of relationships established. I feel that would be harder to specify as an attribute.

### Task 3:

Looking at the code created from my diagram export is creating multiple attributes. The code created from 2.1 is causing all the properties to be set as final. The code created from 2.2. seems to be correct. I would say 2.2 would be the best to export. The code is cleaner and is closer to the original code.

After a few long frustrating hours, I finally have my diagram creating code that does not duplicate attributes. The issue was how the associations were set up. I also had unnecessary associations between classes. I finally realized that I need to make end coming out of the class non navigable.

Leaving navigable or unspecified causes another attribute to be created. I left the generated code in the folder called Export1. I created a new folder named Export1Updated that has the latest export after making those changes. Also below is my update diagram.

