Writing the revised code was a process of thinking of the program as a series of actions as opposed to entities. In the OOP version of the code, data and behavior were encapsulated in the objects, whereas in the functional programming version of the code, the data (e.g the AST tree) was passed explicitly between functions. In the functional programming version, the functions are independent of the specific project, as in they could be used in other programs if needed (functions like extract_imports, check_type_annotations, and extract_docstrings). In terms of immutability, the functional version of the code had variables that were not modified in place whereas they were in the OOP version. In the OOP version of the code, each method was responsible for a combination of responsibilities whereas the functional version split the tasks into smaller pieces. For example, the file_structure method in the OOP version combined counting lines, counting imported packages, counting classes, and counting top level functions. The functional version of the code had a separate function for each of these tasks. Lastly, the functional version of the code used pure functions, like in extract_imports where it analyzed the AST tree and returned a list without altering the tree.