

# OOAD Project Topic Proposal

Nick Cooper, ADD HERE

September 17, 2024

## 1 Hello there!

The section below is some thoughts/ideas to use as a starting point. Feel free to tweak whatever!

## 2 A Quantitative Study of Code Complexity in OO Languages

Our project aims to quantitatively measure the structural complexity of software applications across a range of OO-centric languages. Through this analysis, we hope to detect interesting relationships between language features and end-result programs. As an example, can we formally answer questions like: does multiple inheritance increase or decrease code complexity, on average? There will be two primary technical challenges associated with this project: data, and the quantification of complexity. The first is rather self explanatory; a large dataset of code written in OO languages is critical, as understanding large scale trends will be impossible without a decent sample size. Thankfully, many code datasets have emerged in recent years, thanks to the AI summer. See: Github Code Dataset for an example. The second challenge is far more interesting: how to measure complexity? Because this study will focus on OO languages, we inherit (pun fully intended) some nice theoretical properties right off the bat. Objects and their relations can be easily modeled with many mathematical constructs, such as graphs, categories, and combinatorial complexes, all of which bring their own set of analytical tools. As a simple example, an OO program could be modeled with a graph (possibly a tree), and various notions of complexity explored thereafter: diameter, size of the center, average degree, etc. Many of the aforementioned mathematical structures admit natural notions of “neighborhood”, allowing topological summaries of complexity such as the persistent entropy (see: nice explanation) to be used as well. In summary, we hypothesize that interesting patterns can be extracted by studying the mathematical structure of OO software.