Large Language Models (LLMs) that are pre-trained on code have become the leading method for program synthesis. These models however mainly rely on predicting the next token, which may fail to capture the syntax and meaning of code. Coarse-Tuning Models of Code with Reinforcement Learning Feedback paper hypothesized that teaching the model to provide compilable code by providing compiler feedback will improve performance on downstream tasks. This project expands on that idea with the main difference that compiler feedback is provided to the model in an RL free way, by using methods such as such as Kahneman Tversky Optimization (KTO) and Direct Preference Optimization (DPO). Both methods use feedback to ground the LLM, albeit in slightly different contexts. This thesis seeks to assess the effectiveness of these approaches and their ability to generate compilable code ultimately contributing to advancements, in program synthesis.