PolyPy is a Django web application that utilizes a set of novel, flexible, and easy-to-use question-templating frameworks for generating quasi-random Python coding challenges. It then collects student answers to provide pedagogical insight to CS instructors about both common and individual strengths and weaknesses related to Python coding.

**PROJECT OVERVIEW**

**IMPROVEMENTS**

In this poster, we introduce a set of changes that expand the scope and the pedagogical value of PolyPy:

1) A novel question-templating framework for code-writing challenges
2) Question instance recycling among students for increased efficiency

**FRAMEWORK 2: COMPLETE THE CODE**

This new framework allows the instructor to create a set of unique question instances from a single question skeleton through the selection of "hidden element subsets". PolyPy can now prompt students to complete code segments given a specific objective, drawing more focus than before onto their code-writing abilities.

Instructor creates a code segment with objective

Student requests a "For Loop" challenge

Instructor uses Subset Selection view to choose hidden subsets

PolyPy executes correct code segment & student’s code segment to determine if student is correct

QUESTION RECYCLING

A redesigned business logic brings greater efficiency to the system, enabling the recycling of unique question instances among students so that instructor-designed challenges lead to collections of data that are less prone to high cardinalities, and thus are easier to analyze for patterns in student answers.

Learning Personalization: Create personalized learning paths for students based on coding strengths and weaknesses
Intelli-hints: Build a system that analyzes question content to dynamically provide explanations and hints for solutions
Randomized code completion challenges: allow instructor to bypass the need for manual hidden subset selection; integrate a smart "hidden subset" selector based on detection of keywords & arguments
Pattern Analysis: Implement a system for dynamically aggregating student answers using regex analysis to find common mistakes