

GUPy: Graphics Unplugged (Python)

The primary advantage of teaching beginning programming using GUPy is that students can focus on solving a problem using a specific programming language, but without the frustration and deeply time-consuming process of debugging syntax errors.

GUPy is a programming unplugged activity that helps students practice programming, computational thinking, and problem solving without a computer. The language follows the structure of Python (with a few exceptions), making use of common programming constructs including variables, arithmetic expressions, Boolean expressions, functions, if-statements, and while loops. The outcome of a program is a pattern physically constructed on a grid with tiles of different colors. This mimics computer graphics in which images are created by coloring in pixels within a very large grid (i.e. a graphics window on the computer screen).

MATERIALS

- Grid and Tiles:
 - The Grid consists of a 14x14 grid with labeled columns and rows. It is oriented as one would typically orient the x-y axis when graphing (i.e. column 0 and row 0 are the lower left corner).
 - The Tile page can be printed on 2 different colors of paper and cut along the lines. Note that a single tile takes up 2x2 grid squares. The single grid square that the lower left corner of a tile occupies is its location. For example, a tile placed at column 3 and row 2 is placed over grid squares (3,2), (4,2), (3,3), and (4,3).
- Any game piece or coin can be used to “pin” a location.
- Paper can be used to store variable values.
- The Language (Syntax):
 - The first page of the Language covers the basics constructs.
 - The second page introduces Boolean logic and while loops.
- Coding Examples:
 - Use the Coding Examples to practice READING/TRACING programs.
- Graphics Cards:
- Problem Cards:
 - Problem Cards #1-8: These require only basic constructs of the language to create the pattern (variables are not required).
 - Problem Cards #9-16: These require basic constructs and variables to create the pattern.
 - Problem Cards #30-45: These require variables and if-statements.
 - Problem Cards #50-65: These require the use of a while loop.

Write a GUPy program to hand in:

1. Choose a Problem Card.
2. Choose a Graphics Card (unless the problem card states otherwise).
3. Write a program that when executed will create the pattern presented on the Graphics card.
The color does not always match, but the design should match.

On a blank sheet of lined paper, at the top of the paper, write:

- YOUR NAME(S)
- The # of the Problem Card
- The # of the Graphics Card

Be sure to:

- WRITE A LINE NUMBER to the left of each line of code.
- Write 1 code statement per line.
- Align blocks of code, indenting when appropriate. Note that until you start using if-statements and while loops, every line of code should be left-aligned.

Make sure that you TEST YOUR CODE by TRACING IT. Follow the lines of code EXACTLY as you wrote them and see if the final pattern matches the graphics card. Better yet, hand your code to someone else and have them trace it.