

Review and Practice, Pair Programming

Rose-Hulman Institute of Technology
Computer Science and Software Engineering

Outline

- Review of week 1:
 - Defining and invoking functions
 - Definite loops, using a range statement
 - The Accumulator Loop pattern
- Pair Programming
- Practice, Man. Practice.

Check out project for today

- Go to SVN Repository view, at bottom of the workbench
 - If it is not there,
Window→Show View→Other→SVN → SVN Repositories
- Browse SVN Repository view for **04-ReviewAndPractice** project
- Right-click it, and choose Checkout
 - Accept options as presented
- Expand the **04-ReviewAndPractice** project that appears in Package Explorer (**on the left-hand-side**)

Review in Eclipse

- exampleMainStructure:
 - Starting a program in main.
 - How to define a function, call a function.
- exampleInputComputeOutput:
 - How to organize a program into functions.
 - How to input and convert the input to a float ('int' would convert to an int).
 - How to print (i.e., produce output).
 - How to use a FOR loop.
 - This FOR loop is a definite loop using RANGE.
 - It can be thought of as an example of the Accumulator Loop pattern.
 - How to use local variables
 - for the constant (3.9) and howManyToPrint (10), along with the input variable (x) and the loop variable (k).

Review in Eclipse

- exampleFunctionsWithParameters:
 - Functions with parameters that return values
 - How to define a function with parameters
 - How to call that function, supplying actual arguments whose values the parameters are assigned to
 - Parameters and variables defined in functions are local to that function
 - Capture the returned value in a variable, or use it directly
 - It is all a question of SPECIFICATION - communicating with the user of your code.
 - Appending to a list
- exampleAccumulatorLoopPattern:
 - What the Accumulator Loop pattern is, example

Pair Programming

Becoming a common interview technique!

- Working in pairs on a single computer
 - One person, the *driver*, uses the keyboard
 - The other person, the *navigator*, watches, thinks, and takes notes
- For hard (or new) problems, this technique
 - Reduces number of errors
 - Saves time in the long run
- Works best when partners have similar skill level
- If not, then student with most experience should navigate, while the other student drives.

Q1



Photo by Funchye - <http://flic.kr/p/58xk28>

Find a pair programming partner for HW4

Move to sit next to them for the rest of the period.

Decide who will drive first.

Mmm, food.

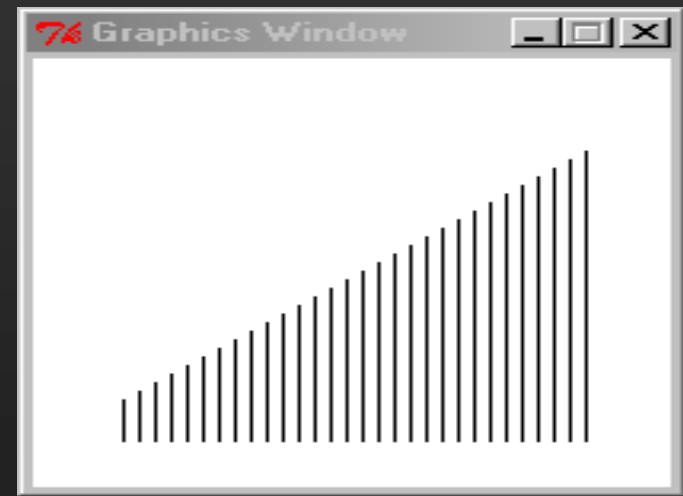
Accumulator Problem: Food Tasting

Accumulating Results: Factorial

- Work with your partner
- Follow the pair programming advice linked from HW4
- Do the TODOs in **factorial.py**

Graphics Exercise with Loops

- Trade pairing roles!
- Do the TODOs in `barChart.py`
- See provided `graphicsExample` module for sample code
 - Consider using variables to hold current x-coordinate and current line length
 - Change the values of those variables each time through the loop



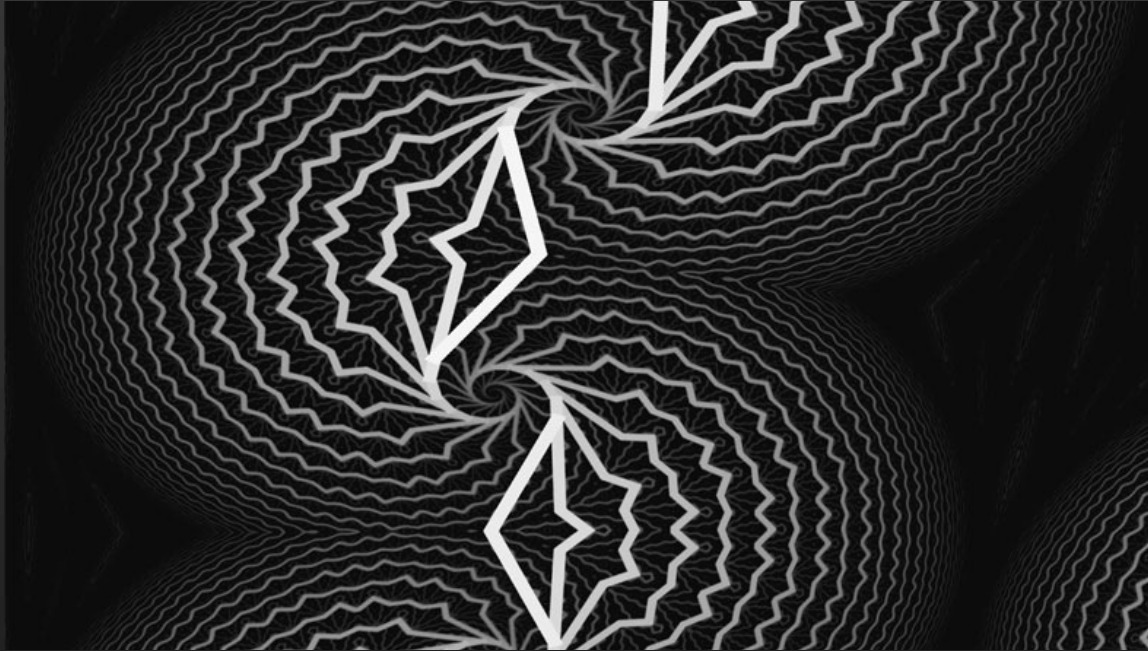


Photo by Sune P - <http://flic.kr/p/5cYBHS>

Feedback, Please

Help me, help you

Q2,3

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Rest of Today: HW4 work

- Pair programming:
 - factorial
 - barChart
- If you finish that, then individual work:
 - bullsEye
 - sumAndCount