Problem 1: Simple Python expressions (10 points)

```
>>> ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
>>> 9 * 8 * (7 // 6 + 5) * 4 + 3 - 2 * 1
1729
>>> 7 < 9 - 5 and 3 % 0 == 3
False
>>> ALPHABET[5:15:3]
'FILO'
>>> ord("F") - ord("A")
5
>>> 
```

The number 1729 is the “taxicab number” that I mentioned in class. The brilliant Indian mathematician Ramanujan instantly recognized 1729 as the smallest integer that could be represented as the sum of two cubes in two different ways ($9^3 + 10^3$ and $1^3 + 12^3$).

Problem 2: Program tracing (10 points)

```
Mystery
64
```

The function $f$ is a recursive implementation of exponentiation in terms of multiplication, which is in turn implemented recursively by $g$ in terms of addition.

Problem 3: Simple Python programs (15 points)

```python
def longestConsecutiveSequence(s):
    
    Returns the longest string of consecutive matching letters in s.
    
    longest = 
    candidate = 
    for ch in s:
        if candidate.startswith(ch):
            candidate += ch
        else:
            candidate = ch
    if len(candidate) > len(longest):
        longest = candidate
    return longest
```
Problem 4: Using the Portable Graphics Library (20 points)

```javascript
# File: RedCross.js

/*
This program solves the practice midterm problem.
*/
rom pgl import GWindow, GCompound, GRect
from gtools import createFilledRect
import random

# Constants
GWINDOW_WIDTH = 800
GWINDOW_HEIGHT = 600
CROSSBAR_LENGTH = 60
CROSSBAR_BREADTH = 20
TIME_STEP = 20
CROSS_SPEED = 2

def RedCross():
    def clickAction(e):
        nonlocal direction
        if gw.getElementAt(e.getX(), e.getY()) == cross:
            direction = random.uniform(0, 360)

    def step():
        cross.movePolar(CROSS_SPEED, direction)

        gw = GWindow(GWINDOW_WIDTH, GWINDOW_HEIGHT)
cross = createRedCross(CROSSBAR_LENGTH, CROSSBAR_BREADTH)
direction = random.uniform(0, 360)
gw.add(cross, gw.getWidth() / 2, gw.getHeight() / 2)
gw.addEventListener("click", clickAction)
timer = gw.createTimer(step, TIME_STEP)
timer.setRepeats(True)
timer.start()

def createRedCross(length, breadth):
    """
    Creates a GCompound consisting of a red cross centered at the origin. The
    parameters length and breadth specify the larger and smaller dimension of
    the rectangles forming the cross, respectively.
    """

cross = GCompound()
horizontalBar = createFilledRect(-length / 2, -breadth / 2,
    length, breadth, "Red")
verticalBar = createFilledRect(-breadth / 2, -length / 2,
    breadth, length, "Red")
cross.add(horizontalBar)
cross.add(verticallBar)
return cross

# Startup code
if __name__ == "__main__":
    RedCross()
```

Please remember that the midterm is open-book.
Friday, October 11, 3:10–4:30 P.M., PSYCH 105