

Solutions to Practice Midterm #2

Problem 1: Simple Python expressions (10 points)

```
IDLE
>>> ALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
>>> 1 * 2 % 3 + (4 % 5 * 6) * 7 * 8 // 9 * 10
1492
>>> str(5 * 2) + str(5 + 1) * 2
1066
>>> ALPHABET[-9::-13] + ALPHABET[4:2:-1]
REED
>>> ALPHABET[len(ALPHABET)]
Traceback (most recent call last):
  File "<stdin>", line 11, in <module>
IndexError: string index out of range
>>>
```

The number 1492 is, of course, the year in which Columbus reached the Americas; 1066 is the year in which William defeated Harold at the Battle of Hastings.

Problem 2: Program tracing (10 points)

```
Mystery
b
bagg
baggage
```

Problem 3: Simple Python programs (15 points)

```
# File: Spoonerize.py
"""
This module defines a function that creates spoonerisms.
"""

from PigLatin import findFirstVowel

def spoonerize(phrase):
    """
    Creates a spoonerism by swapping the leading consonant strings
    from the first and last words in the phrase.
    """
    firstSpace = phrase.find(' ')
    lastSpace = phrase.rfind(' ')
    firstWord = phrase[:firstSpace]
    lastWord = phrase[lastSpace + 1:]
    middle = phrase[firstSpace:lastSpace + 1]
    newFirstWord = getHead(lastWord) + getTail(firstWord)
    newLastWord = getHead(firstWord) + getTail(lastWord)
    return newFirstWord + middle + newLastWord

def getHead(word):
    return word[:findFirstVowel(word)]

def getTail(word):
    return word[findFirstVowel(word):]
```

Problem 4: Using the Portable Graphics Library (20 points)

```

# File: Fireworks.js

"""
This program simulates a firework launch.
"""

from pgl import GWindow, GOval
from RandomCircles import randomColor
import random

# Constants

GWINDOW_WIDTH = 500;      # pixels
GWINDOW_HEIGHT = 300;    # pixels
DELTA_RADIUS = 2;        # pixels
TIME_STEP = 20;          # milliseconds
FLIGHT_TIME = 1200;      # milliseconds
EXPANSION_TIME = 500;    # milliseconds

def Fireworks():
    def step():
        nonlocal r, t
        if t > FLIGHT_TIME + EXPANSION_TIME:
            timer.stop()
        elif t > FLIGHT_TIME:
            r += DELTA_RADIUS
            firework.setBounds(firework.getX() - DELTA_RADIUS,
                               firework.getY() - DELTA_RADIUS,
                               2 * r, 2 * r)
        else:
            firework.move(dx, dy);
        t += TIME_STEP

    random.seed()
    gw = GWindow(GWINDOW_WIDTH, GWINDOW_HEIGHT)
    r = 1
    firework = GOval(gw.getWidth() / 2, gw.getHeight(), r, r);
    firework.setColor(randomColor())
    gw.add(firework)
    targetX = random.uniform(0, gw.getWidth())
    targetY = random.uniform(0, gw.getHeight() / 2)
    nSteps = FLIGHT_TIME / TIME_STEP
    dx = (targetX - firework.getX()) / nSteps
    dy = (targetY - firework.getY()) / nSteps
    t = 0
    timer = gw.createTimer(step, TIME_STEP)
    timer.setRepeats(True)
    timer.start()

# Startup code

if __name__ == "__main__":
    Fireworks()

```