SCIE1000, Computing Tutorial Week 11, Semester 1, 2011.

# Aims

The aims of this computing tutorial are consolidate your programming skills, by modifying an existing program to solve a related problem.

1. On Page 267 of the lecture notes, there is a program to use Riemann sums or the trapezoid rule to estimate the area under a nicotine concentration curve. A copy of this program, called *AUC.py*, is available on the Blackboard site, in the *Learning materials/Tutorial materials* section. Download a copy of the program and modify it so that it:

   - **only** uses the trapezoid rule to estimate the AUC. The program should still plot a graph showing the indivifual trapezoids used to estimate the area.
   - uses the measured increased blood glucose concentrations given in Question 1(b) of the **tutorial sheet** this week, rather than the blood nicotine levels that it currently uses.
   - prompts the user to enter the values of $a$, $b$ and $c$ for a quadratic equation $at^2 + bt + c$ that models the increased blood glucose levels after consuming a controlled dose of pure glucose, and then uses the Fundamental Theorem of Calculcus to find $\int_0^2 g(t)\,dt$. Your program should also plot a graph of $g(t)$ between $t = 0$ and $t = 2$, on the same axes as the graph of the trapezoids.
   - estimates the GI of the food, in the same way as you did by hand in Question 1(d) of the **tutorial sheet** this week.

2. ## Assessment Question (1.5%): GI program.

   Ensure that your program from the previous question is commented, has sensible variable names, includes useful output messages, and has appropriate formatting of the graph. Use your program to estimate the GI of the food from Question 1 of the tutorial sheet. Compare your answers with those you obtained during the tutorial, and explain any differences. Which answer is likely to be more accurate?

   This program will contribute 1.5% towards your final assessment for SCIE1000. **To receive these marks, you must demonstrate your program and its output to a tutor during your tutorial.** Program output, including text messages and correct values, are together worth up to 1.0%. Appropriate comments and variable names are together worth 0.5%.

3. (Final exam, 2010. Worth 7 marks, so about 7 minutes to work.) Write all of the output generated by the following Python program **when the user enters the value 7 from the keyboard**.

```
from __future__ import division
from pylab import *
# The command x % y gives the remainder when x is divided by y.
#
n = input("Enter value? ")
a = zeros(n)
a[1] = 1
i = 2
while i<n:
    a[i] = a[i-1] + a[i-2]
    i = i + 1
print "a: ",a
b = a % 2
print "b: ",b
```

**The end**