Lab 1 – Monday, September 11, 2006

Write Python programs to solve the following problems:

1. The Konditorei coffee shop sells coffee at $10.50 a pound plus the cost of shipping. Each order ships for $0.86 per pound + $1.50 fixed cost for overhead. Write a program called **`coffee()`** that calculates the cost of an order. For example:
```
>>> coffee()
How many pounds of coffee would you like? 7
The total cost of your order is $ 81.02
```

2. Write **`addFirst(n)`**, which adds up the first *n* numbers starting from 1.
```
>>> addFirst(100)
The sum of the numbers from 1 to 100 is 5050
```

3. Write **`addSquares(n)`**, which adds up the first *n* squares starting from 1.
```
>>> addSquares(10)
The sum of the squares of the numbers from 1 to 10 is 385
```

4. Write **`addCubes(n)`**, which adds up the first *n* cubes starting from 1.
```
>>> addCubes(10)
The sum of the cubes of the numbers from 1 to 10 is 3025
```

5. Write **`addSeries(n)`**, which adds up *n* numbers entered by the user.
```
>>> addSeries(3)
Enter next number: 7
Enter next number: -3
Enter next number: 12
The sum of the 3 numbers you entered is 16
```

6. Write **`average(n)`**, which averages *n* numbers entered by the user.
```
>>> average(3)
Enter next number: 7
Enter next number: -3
Enter next number: 12
The average of the 3 numbers you entered is 5.333333333333333
```

7. Write **`addOdds(a, b)`**, which adds the odd numbers in the range *a* to *b*, inclusive. You may assume that *a* will be an odd number (but *b* could be even or odd).
```
>>> addOdds(1, 5)
The sum of the odd numbers from 1 to 5 is 9
>>> addOdds(11, 15)
The sum of the odd numbers from 11 to 15 is 39
>>> addOdds(11, 16)
The sum of the odd numbers from 11 to 16 is 39
```

(continued on back)

8. Write a program called **`approx()`** that approximates the value of π by summing the terms of this series:

$$\pi = 4/1 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + ...$$

Your program should prompt the user for *n*, the number of terms to sum, and then output the sum of the first *n* terms of this series. Have your program print out a comparison of its approximation to the value of `math.pi` to see how accurate it is.

9. A Fibonacci sequence is a sequence of numbers where each successive number is the sum of the previous two. The classic Fibonacci sequence begins: 1, 1, 2, 3, 5, 8, 13, 21, ... Write a program that computes the *n*th Fibonacci number:

```
>>> fib(1)
1
>>> fib(6)
8
```