Life is hard.
Computing helps lots.
It works how?
Introduction: Why Bother?

- Computations solve problems.
- Some computations fast, *e.g.*, Google search.
- Some seem hard, *e.g.*, getting good class schedules, or we hope they are, *e.g.*, cracking encrypted communications.

**How do we solve problems quickly?**

**How do we show problems are hard?**

**How do we deal with hard problems?**
Problem: A set of inputs and their associated outputs.

Algorithm: A sequence of instructions that solves a problem, i.e., computes the output for a given input.

Program: A sequence of instructions in some computer language that solves a problem.
Finding the Area of a Circle

Problem:

**Input**: A radius $r$.

**Output**: The area of a circle with radius $r$.

Algorithm:

$$\text{area} = 3.14159 \times r \times r$$

print area

Program:

```python
import sys
r = sys.argv[1]
area = 3.14159 * r * r
print area
```
Summing a List

Problem:

**Input:** A list $L$ of $n$ numbers.
**Output:** The sum of the numbers in $L$.

Algorithm:

```
sum = 0
for i = 1 to n do
    sum = sum + L[i]
print sum
```

Program:

```
sum = 0
for i in range(1, n + 1):
    sum = sum + L[i]
print sum
```
Searching a List

Problem:

**Input:** A list $L$ of $n$ elements and a value $t$.
**Output:** The position of the element in $L$ with value $t$ if such an element exists and $-1$ otherwise.

Algorithm / Program:

```python
tpos = -1
i = 1
while (i <= n) and (tpos == -1):
    if L[i] == t:
        if L[i] == t:
            tpos = i
        i = i + 1
print tpos
```
Solving Problems with Programs: The Big Picture

Algorithm

Code

Editor / Word Processor

Program

Compile

Compiler

Executable

Run

Operating System

Results
Science 1000: Lecture #1 (Wareham):

Under the Hood:
Programs, Algorithms, and Problems

Life is hard.
Computing helps lots.
It works how?