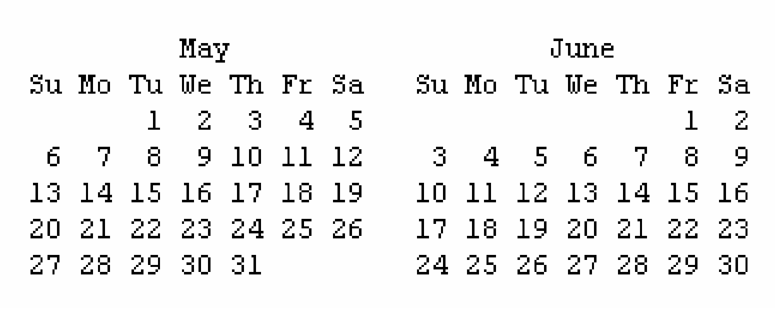
A201/A597 Introduction to Programming I

`http://www.cs.indiana.edu/classes/a201-dger`

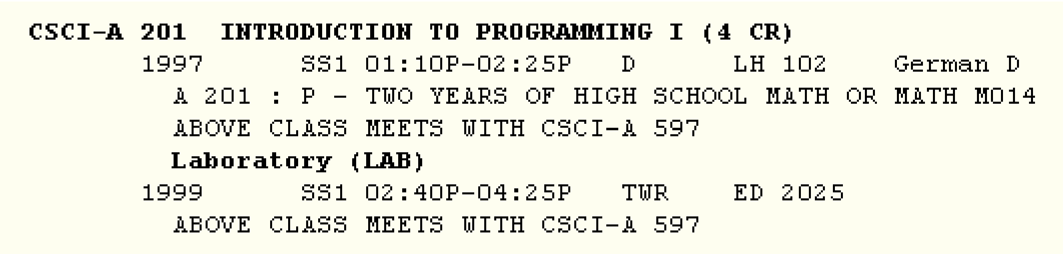
First Summer 2007

Classes start: Tuesday, May 8
Last day of classes: Thursday, June 14



```
         May                         June
Su Mo Tu We Th Fr Sa     Su Mo Tu We Th Fr Sa
       1  2  3  4  5                    1  2
 6  7  8  9 10 11 12      3  4  5  6  7  8  9
13 14 15 16 17 18 19     10 11 12 13 14 15 16
20 21 22 23 24 25 26     17 18 19 20 21 22 23
27 28 29 30 31           24 25 26 27 28 29 30
```

Thus the class spans only 6 weeks (summer session is usually very intense.)



```
CSCI-A 201  INTRODUCTION TO PROGRAMMING I (4 CR)
       1997      SS1 01:10P-02:25P   D      LH 102    German D
         A 201 : P - TWO YEARS OF HIGH SCHOOL MATH OR MATH M014
         ABOVE CLASS MEETS WITH CSCI-A 597
         Laboratory (LAB)
       1999      SS1 02:40P-04:25P   TWR    ED 2025
         ABOVE CLASS MEETS WITH CSCI-A 597
```

Lecture will meet in LH (Lindley) 102 and will be taught by Adrian German

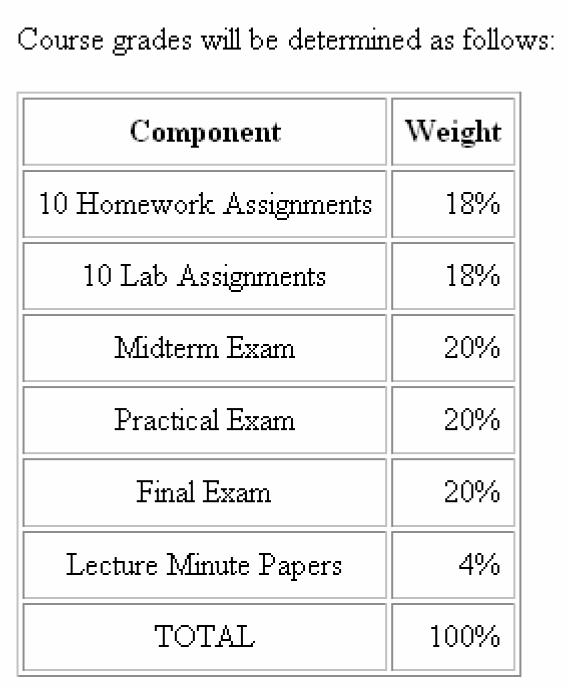
Adrian German

Office: Lindley 201D
Phone: 812 855-7860
E-mail: dgerman@indiana.edu
Office hours: 11am-noon daily (MTWRF) or by appt.

Lecture in LH102 daily (MTWRF) at 1:10-2:25pm

No lecture on Monday May 28 (Memorial Day)

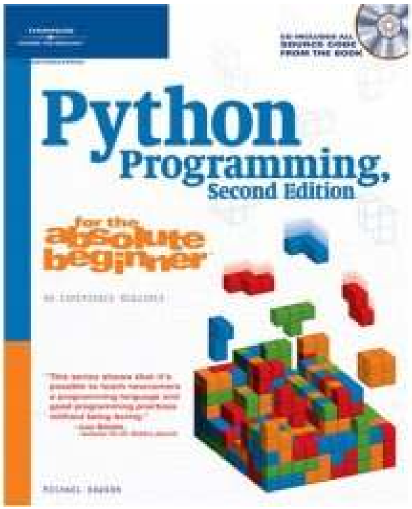
There will be 25 lectures plus a midterm (5/31) and a final (6/14).
There will be 16 labs plus a practical (6/5) and a wrap-up lab (6/14).



Course grades will be determined as follows:

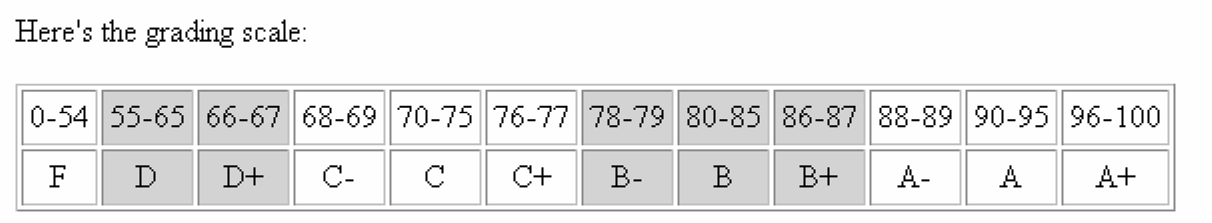
Component	Weight
10 Homework Assignments	18%
10 Lab Assignments	18%
Midterm Exam	20%
Practical Exam	20%
Final Exam	20%
Lecture Minute Papers	4%
TOTAL	100%

This is the text required for this class:



You can read[1] it for free in Books24x7.

Exams (both the two written exams and the practical) will be closed book and based entirely on the homework and the lab assignments covered up to that point.



Here's the grading scale:

0-54	55-65	66-67	68-69	70-75	76-77	78-79	80-85	86-87	88-89	90-95	96-100
F	D	D+	C-	C	C+	B-	B	B+	A-	A	A+

Labs will be taught by Metal (Nathaniel) Hayes:

Metal (Nat) Hayes

Office: Lindley 201D (for the whole summer).

E-mail: nahayes@indiana.edu

Labs in ED025 TWR 2:40-4:25pm

Metal is cool. You will like him.

Metal digs Blender 3D.

[1] http://www.libraries.iub.edu/scripts/countResources.php?resourceId=59555

Lecture One: Tuesday May 8, 2007 (Lindley Hall 102)

What is a computer program?

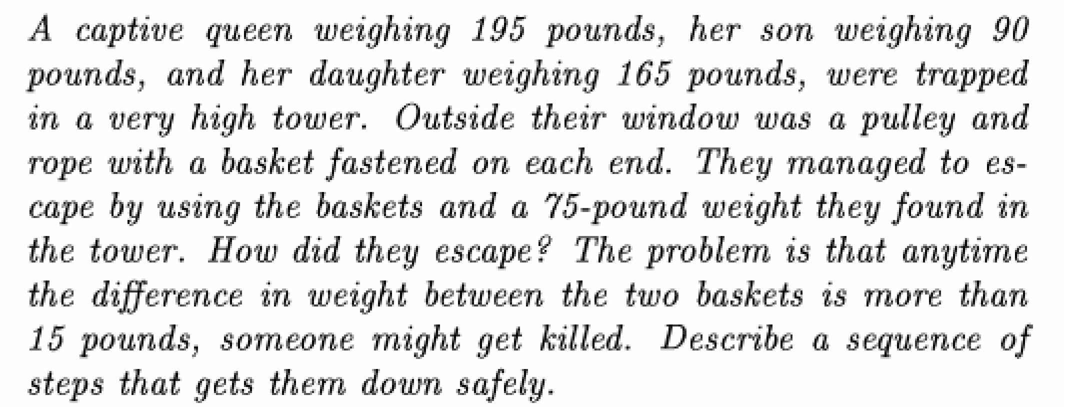
A computer program is a set of instructions that a computer follows to perform a task.

What is a computer?

A computer is a device that follow instructions.

What is computer programming like?

Try this:



A captive queen weighing 195 pounds, her son weighing 90 pounds, and her daughter weighing 165 pounds, were trapped in a very high tower. Outside their window was a pulley and rope with a basket fastened on each end. They managed to escape by using the baskets and a 75-pound weight they found in the tower. How did they escape? The problem is that anytime the difference in weight between the two baskets is more than 15 pounds, someone might get killed. Describe a sequence of steps that gets them down safely.

This is our problem for today's lecture.

We will see that most of the times:

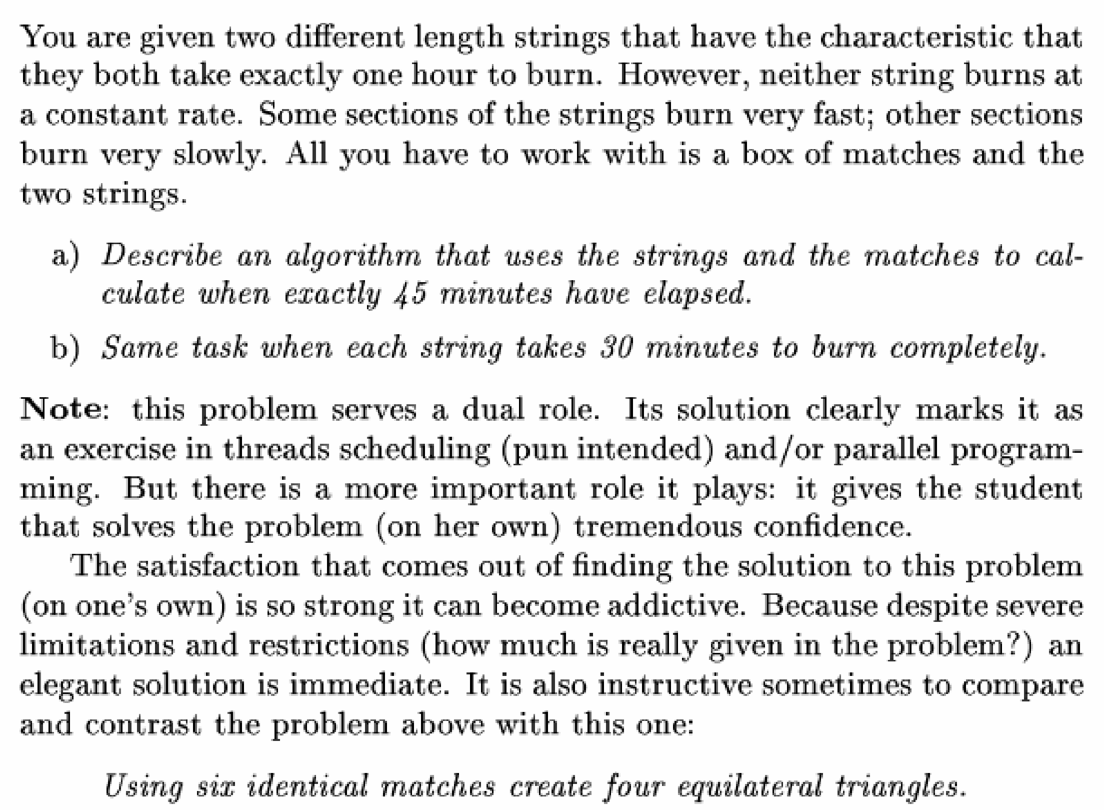
- we need to write a program for a problem we don't know how to solve
- so we need to find a sequence of steps that solves the problem (algorithm)
- prove that the algorithm is correct (it correctly solves the problem), and
- implement (code) the algorithm in a programming language

Programming language: language that the computer understands.

In this class we will study/learn the Python programming language.

Today in lab you will implement a program for the captive queen problem. You will be using a system called Alice that makes use of its own programming language. We will explore some of the features of Alice first then proceed with the implementation.

Exercises to think about for tomorrow's lecture:



You are given two different length strings that have the characteristic that they both take exactly one hour to burn. However, neither string burns at a constant rate. Some sections of the strings burn very fast; other sections burn very slowly. All you have to work with is a box of matches and the two strings.

a) *Describe an algorithm that uses the strings and the matches to calculate when exactly 45 minutes have elapsed.*

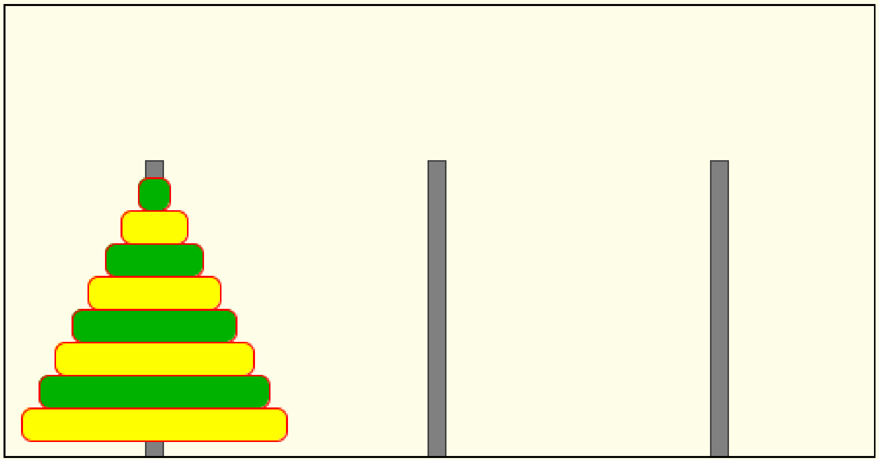
b) *Same task when each string takes 30 minutes to burn completely.*

Note: this problem serves a dual role. Its solution clearly marks it as an exercise in threads scheduling (pun intended) and/or parallel programming. But there is a more important role it plays: it gives the student that solves the problem (on her own) tremendous confidence.

The satisfaction that comes out of finding the solution to this problem (on one's own) is so strong it can become addictive. Because despite severe limitations and restrictions (how much is really given in the problem?) an elegant solution is immediate. It is also instructive sometimes to compare and contrast the problem above with this one:

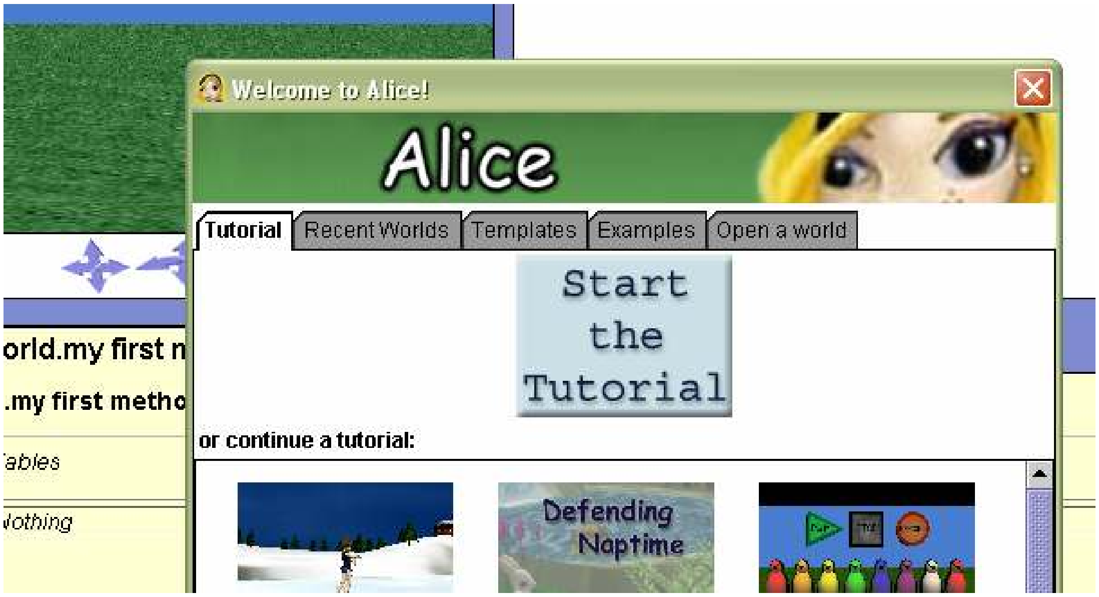
Using six identical matches create four equilateral triangles.

Also: The Tower of Hanoi puzzle, invented by the French mathematician Edouard Lucas in 1883. We are given a tower of eight disks (initially four in the applet below), initially stacked in increasing size on one of three pegs. The objective is to transfer the entire tower to one of the other pegs (the rightmost one in the applet below), moving only one disk at a time and never a larger one onto a smaller.



Lab One: Tuesday May 8, 2007 (Education Building, room 2025)

1. Start Alice. Run the tutorial.



While you do this Metal will come to write down your name.

2. Start a new world, add a chicken, program the chicken to turn left, move forward one meter, turn to face camera and play the sound of a chicken while thinking (for 8 seconds) some thought (like “Boy, is this interesting or what?!”).

Show this to Metal when you’re done.

3. Implement an algorithm for the Captive Queen problem in Alice.

Show Metal your implementation at the end of the lab. It’s your lab assignment for today.

We’ll develop the code together in lab.