Introduction to:
Computers & Programming:
Planning a Complex Program

Adam Meyers
New York University
How to Solve a Complex Problem

• Break down big problem into small problems
  – For each small problem either:
    • Solve the problem if you know how
    • Break it down further if you don't know how to solve it and:
      – Try to solve each of the smaller problems
      – Etc.

• We will ignore our current limitations in Python: We can implement this later in the term
Until we find problems easy enough to solve.
Exercise – Let's Write an Algorithm for a Complex Problem

• Given
  – Today's date:
    • Month, day, year, and day of the week
  – An arbitrary number of days in the future (e.g., 1500)

• How should we go about identifying that day in the future?
  – Month, day, year and day of the week
How Can We Solve This Problem?

1. Some Pieces of the Problem
   a) Find the year
   b) Find the month
   c) Find the day
   d) Find the day of the week

2. Are all calculation dependent on each other or are any independent of the others, i.e., is there an order

3. Write Out Pseudo Code for Solving This Problem
The Problem 1: Calculate day of week

- Day_of_Week_table 0-6 --> Monday, Tues, Wed, Thur, Fri, Sat, Sun
- Map current_day_of_week to number in table
- MOD = (current_day_of_week + total_num_of_days) mod 7
- future_day_of_week = lookup MOD in Day_ofWeek_table
Calculate Year in Future

• Day_of_Year = 
  Calculate_day_of_year(month, day, leap_yes_or_no)

• years_in_future = 
  complex_int_div(day_of_year + days_in_future, leap_yes_or_no)
Calculate_day_of_year

- `table_of_days_per_month` with 2 entries for February
- Add up days in `previous_months(current_month)`
  - `make_table_from 0,11 → Jan,…,Dec`
  - `current_month_num = lookup(current_month)`
  - For each month from 0 to `(current_month-1)`:  
    - `add_in_number_from table_of_days to day_of_year`
  - Add day of month to `day_of_year`
Month in Future

• Go thru days in month starting with Jan
  – Keep subtracting from day_in_future_year
    • Until the remainder is lower than the days in that month – that is the month_in_future
    • The remainder is the day of the month (in future)