More on Python, Tools, Compsci 101

- APTs, Assignments, Tools
  - APT: Algorithmic Problem-solving and Testing
  - How to get help and when to get it
- Types, values, and operations on them
  - Names and types: int, float, string, file, list, long, …
  - Operations: *, +, -, /, %, **
  - Functions and methods on types: string, list, files
- Functions: organizing code (and ideas)
  - Functions, modules, indentation, naming
  - Parameterization to generalize solutions
  - Return values to caller of function

Functions: abstractions over code

- Naming something gives you power
  - How do you read a file into a string?
  - What is length of a string? Of a list?
- We can write and call functions
  - Re-use and/or modify
  - Store in module, import and re-use functions
  - Import standard modules and use functions from them
- Functions can (should?) return a value
  - We've seen len return an int, what about file.read()?
  - Other functions return Strings, floats, or other types

Re-use: Counting words in file

```
def wordCount(filename):
    file = open(filename)
    all = file.read()
    words = all.split()
    return len(words)
if __name__ == "__main__":
  name = "/data/romeo.txt"
  print "# words in",name,
  print "=",wordCount(filename)
```

Running Python Program/Module

- Python is an interpreter, platform specific
  - So is Java, so is Android, … contrast compilers
  - Python can execute a .py file, need a "launch point"
- Convention in Python and other languages
  - Start with section labeled __main__, that's run

```
if __name__ == "__main__":
    statements here
    Statements here
```

- Boilerplate, don't memorize

Anatomy of a Python function

```
def name(params):
    body
```

- Define a function, provide a name, provide parameters, provide a function body
  - How to decide on name?
  - Do we need parameters?
  - What does body of function do
- Functions provide a named abstraction over code
  - Huh? What is `math.factorial(15)` `"hello".upper()`

Revisiting functions

- Python Heron's formula or BMI (Body Mass Index)
  - What's the name of the function
  - What are parameters that enable function use/call
- How do we write and test APTs
  - What's the name of the function
  - What are parameters that enable function use/call
  - Who writes the function? Who calls the function?
- How will you decide on these things in writing your own code?

Design, Implementation, Testing

- Designing Python code and functions
  - What do you want the code to do?
  - Other aspects of code, e.g., portability, efficiency, size, …
  - Understand how to solve a problem without computer
- Implementing design in code
  - Translation of ideas into vocabulary of Python
  - We don't have a large vocabulary, but it will grow!
- Testing code, functions
  - How do you know when function is right?
  - What confidence can testing provide?
  - APT testing is similar to Unit Testing (well known)

Nancy Leveson: Software Safety

- Mathematical and engineering aspects, invented the discipline
  - Air traffic control
  - Microsoft word

  "There will always be another software bug; never trust human life solely on software" huffington post?

- Therac 25: Radiation machine
  - http://en.wikipedia.org/wiki/Therac-25
  - http://bit.ly/5qOjoH
- Software and steam engines

Running Python

- Can run in Eclipse Console Window
  - How to start? What to type?
  - Also run on command-line, e.g. simple Mac/Linux
- Can import code into another module/.py file
  - See APT examples and how to run them
  - Understand how your Python code is executed
  - Understand where Python code is and how it got there
- How do we test your code to grade it, evaluate it?
  - APTs: auto-test, other assignments, human-in-the-loop

Language and Problems in Context

- Convert Spanish Wikipedia page to English
  - How do we convert HTML to text?
- How do you determine if 2040 is a leap year?
  - Any specified year is a leap year?
- How do we make an image larger, more red, …
  - What is an image? How do read it? Convert it? Access it?
- How do we find the BMI for everyone
  - What's the right tool for this? Why use Python? Why not?

Looping in Python

- What is a string?
  - *Sequence* of characters that supports some operations
- What is a list?
  - *Sequence* of elements that supports some operations
- What is a (text) file?
  - *Sequence* of lines that supports some operations
- Elements of the sequence are indexed
  - What is an index, where does indexing begin? Zero!
  - Value of `s[a]`, `s[a:b]`, `s[a:]` compare `s[:b]`

How do we loop over a sequence?

```
for x in seq:
    do something to x
    do another thing
```

- Loops are powerful
  - Format? Syntax?
  - Errors?
- Loops in Lightbot?
  - Implicit
- Loops in string.split(), loops in file.read()
  - Again implicit: Advantages? Disadvantages?
- What about conditional execution? Next week!