Act like a code monkey

Coding Basics
Announcement

- Homework 1 grade is posted
- If you believe there is an error in grading (assignments or quizzes), you may request a regrading within one week of receiving your grade. Requests must be made in writing, explaining clearly to the TA why you think your solution is correct.
Base-3 System
Base-3
0, 1, 2
Decimal

\[
100 = 10^2 \quad 10 = 10^1 \quad 1 = 10^0
\]

\[
1 \times (10^2) + 2 \times (10^1) + 5 \times (10^0) = 125
\]
Decimal

\[ 9 \times (10^2) + 7 \times (10^1) + 6 \times (10^0) = 976 \]
Decimal

\[ 3 \times (10^2) + 4 \times (10^1) + 5 \times (10^0) = 345 \]
Decimal

\[2 \times (10^2) + 1 \times (10^1) + 1 \times (10^0) = 211\]
base-3

$9 = 3^2$  \hspace{1cm} $3 = 3^1$  \hspace{1cm} $1 = 3^0$

$2 \times (3^2) + 1 \times (3^1) + 1 \times (3^0) = 22$ (decimal)
base-3

9 = 3^2  3 = 3^1  1 = 3^0

2x(3^2) + 2x(3^1) + 0x(3^0) = 24 (decimal)
base-3

\[ 1 \times (3^2) + 0 \times (3^1) + 2 \times (3^0) = 11 \text{ (decimal)} \]
Base-4 System
Base-4
0, 1, 2, 3
base-4

\[
1 \times (4^2) + 0 \times (4^1) + 2 \times (4^0) = 18 \text{ (decimal)}
\]
Think beyond Binary
Quantum Computing

- Quantum computers: use quantum-mechanical phenomena to perform operations on data
- Different from digital electronic computers based on transistors.
- Uses quantum bits (qubits), which can be in superpositions of states: e.g. linear combination of basic states of particles
- Quantum superposition: any 2+ quantum states can be added together and the result will be another valid quantum state
- Quantum Turing machine
- Non-deterministic and probabilistic
- Paul Benioff, Yuri Manin 1980; Richard Feynman 1982; David Deutsch 1985.
- Further reading: https://en.wikipedia.org/wiki/Quantum_computing
A quantum bit corresponds to a **single electron in a particular state**. Using the trajectories of an electron through two closely spaced channels for encoding.

In principle, 2 different states are possible: the electron either moves in the upper channel or in the lower channel – a binary system.

However, a particle can be **in several states simultaneously**, that is, it can quasi fly through both channels at the same time.

These overlapping states can form an extensive alphabet of data processing.

**Quantum computer science**

Further reading: [http://qist.lanl.gov/qcomp_map.shtml](http://qist.lanl.gov/qcomp_map.shtml)

Problem Solving
Problem solving

1. Formulate problem
2. Think creatively about solutions
3. Express a solution clearly & accurately
Learn to program

= Learn to solve problems
Python Programming Language

Credit: lecture notes modeled after http://www.openbookproject.net/thinkcs/python/english2e/index.html
Python

- high-level language, like C++, JAVA
- Different from low-level language like assembly language that has strong relation to machine code
- Easier, more efficient to write
- More likely to be correct, portable
Typically considered as an interpreted language

Source code → Interpreter → Output

read line by line
perform computation
Shell Mode in Python Shell

$ python
Python 2.7.1 (r271:86832, Jun 16 2011, 16:59:05)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print 1+2
3
$ python myprogram.py
3

Script Mode

Used primarily for this class
for programs more than a
few lines

myprogram.py

print 1+2
We use online Python interpreters

Making life easier

http://www.tutorialspoint.com/execute_python_online.php
http://www.skulpt.org/
http://www.pythontutor.com/index.html

https://repl.it/languages/python3 (note, this is Python 3, some syntax rule is more strict)
Coding Basics
What is a program?
A sequence of instructions that specifies how to perform a computation.
Basic Instructions of a program
Input: get data from the keyboard, a file, or some other device
**Output**: display data on the screen or send data to a file or other device
Math: perform basic math operations like additions and multiplication
Conditional execution: check for certain conditions and execute the appropriate sequence of statements
Repetition: perform some action repeatedly, usually with some variation
That's all need to know about a program!
programming
≡ problem solving
Breaking a large, complex task to smaller and smaller subtasks, until the subtasks are simple enough to be performed by some basic instructions...
What is debugging?
Grace Hopper
First actual case of bug (moth)
“Amazing Grace”
bug = programming error  
debugging = track down the bugs and fix them
Type of bugs

- **Syntax error**: violation of rules and structures. For example, a sentence has to start with a capital letter...
- **Runtime error**: error does not appear until program is executed (rare for now)
- **Semantic error**: the program is not doing what you tell it to do. Tricky to track down.
programming = experimental debugging, detective work, hypothesis testing, etc.
When you have eliminated the impossible, whatever remains, however improbable, must be the truth.

-- Sherlock Holmes
Debugging can be intellectually challenging and rewarding.
Python is a formal language used to express computations.
Tokens: words, numbers, etc.
Statements: arrangements or structures of tokens
1st Python Lab

Credit: https://pixabay.com/static/uploads/photo/2014/04/02/14/05/snake-306109_960_720.png
print "Hello World!!!"
Hello World!!!
print "What happened to Han Solo?!"
What happened to Han Solo?!
print “My name is Harrison Ford.”

your name goes here!
My name is Harrison Ford.
print Star Wars
File "<stdin>", line 1
print Star Wars
^  
SyntaxError: invalid syntax

- Runtime Error
- Need quotation marks “ ” or ‘ ’
Values and Data Types
A value is something a program manipulates

- a letter “a”
- a number 2
- a sentence “hello world!”
There are different types of values

- 2 is an integer
- “Hello World” is a string
- 3.1416926 is a floating point (real # or approximation of real #)
2nd Python Lab
print 5
print 100000
1,000,000
print 100,000
Treating 100,000 as a list of 2 items
type("hello world!")
<type 'str'>
type(2)
<type 'int'>
type("2")
<type 'str'>
type(3.1415926)
<type 'float'>
What is a variable?
A name that refers to a value
Assignment statement: creates new variables and gives them values
message = “Orange is the new black.”
n = 28
\pi = 3.1416926
print message
print n
print pi
Orange is the new black.
28
3.1415926
type(message)
type(n)
type(pi)
<type 'str'>
<type 'int'>
<type 'float'>
28 = n
File "<stdin>", line 1
SyntaxError: can't assign to literal
Choose meaningful variable names

- What = “hello” (not so good)
- Begin with letters, contain letters and numbers and “_”
- Typically use lowercase letters
- Python keywords (describe rules and structures) can’t be variables
3nd Python Lab
name = "Tom"
File "<stdin>", line 1
321name = "Tom"
^
SyntaxError: invalid syntax
ineedmoney$ = 100
File "<stdin>", line 1
ineedmone$ = 100
^ SyntaxError: invalid syntax
class = 1999
File "<stdin>", line 1
class = 1999
^
SyntaxError: invalid syntax
What is a statement?
An instruction that the Python interpreter can execute
Examples:
print statement
assignment statement
Python executes a statement and display the results (if any)
Result of a print statement is a value
Assignment statement produces no output
A script/program contains a set of statements, results appear one at a time.
print "a"

x = 28

print x
a
28
What is an expression?
A combination of values, variables and operators
Evaluating an expression by an interpreter produces a value
Shell Mode (command line)

>>> 1+1
2
>>> 28
28
message = 'what is up?'
message
'what is up?'
print message
what is up?

print statement
print values
What is an operator?
Operators are special symbols representing computations, such as addition and multiplication.
Operators use operands
+, -, *, / (integer division)
** exponentiation
11+22
hour*60+minute
minute/60
3**2+3**1
(3**2)+(3**1)
Order of operations

()`

**

* , /
2**2 + 1  # (2**2) + 1
3 * 1**3  # 3 * (1**3)

#comments
>>> 2**2+1
5

>>> 3*1**3
3
Operations on Strings
>>> fruit = "apple"
>>> bakedgood = "pie"
>>> print fruit + " " + bakedgood
apple pie
Input
>>> n = input("Enter a numerical expression ")
Enter a numerical expression 1+3
>>> print n
4
```python
>>> n = raw_input("Enter a numerical expression ")
Enter a numerical expression 1+3
>>> print n
1+3
```

`raw_input()` is replaced by `input()` for Python 3.*
Combination of statements
>>> print "3+2+1", "is equal to ", 6
3+2+1 is equal to 6
>>>
Play with Python labs on your own!
THANKS!

Any questions?

You can find me at beiwang@sci.utah.edu

http://www.sci.utah.edu/~beiwang/teaching/cs1060.html
Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash