Penguin Honoured by Queen

London (Antarctic News Service)—A revered Scottish citizen of the penguin persuasion was knighted today by Queen Elizabeth II. “This bold and brave bird has backed British battalions in both Bolivia and Bangladesh,” she announced. “Bruce Burns has blown on his bagpipes in barracks and barns, brought blessings to bombardiers, and basically been the best brother a Briton could beg for.

Penguin Burns was modest in his acceptance of the honour. “I appreciate the Queen’s approbation,” he commented. “All of this adulation and these awards have been acquired because of my awesome army of associates. I always appreciate their amazing additions to my academic accomplishments, and they always accept my accent. Accolades to all!”
What We Hope You’ve Learned

Big picture: CS is broad and rich

- Artificial intelligence
- Proofs of uncomputability
- Robotics
- Graphics
- User Interfaces
- NLP
- Big Data Ethics
- …and lots more!
All About the Final

• 3 hours long
• Monday, December 17th, 2-5 PM
• Here in Beckman B126
• Bring two 8.5x11” sheets, double-sided
  • Contents are up to you
  • Hmmm reference will be provided
What does `cmp(x)` return for each of these integers, `x`?

\[
\text{cmp}(100\ldots000) = 16 + \frac{11}{\text{1 googol zeros} \text{ a googolplex}} = 27
\]

\[
10^{10^{100}}
\]

What does `cmp(x)` return for each of these strings?

- arithmetic **
- lastweekofclass **
- notenoughcoffee = ?

Bug-Finding Person! (or Pony...)

https://www.deviantart.com/synthrid/
What does `cmp(x)` return for each of these integers, `x`?

\[
\text{cmp}(100...000) = 16 + \frac{11}{1 \text{ googol zeros} \cdot \text{a googolplex}} = 27
\]

\[
10^{10^{100}} = 10^{(10^{100})}
\]

What does `cmp(x)` return for each of these integers, `x`?

- arithmetic **
- lastweekofclass **
- notenoughcoffee = ?

Bug-Finding Person!
(or Pony...)

https://www.deviantart.com/synthrid/
What Will Be Covered?

• Comprehensive exam (entire semester)
• Some things are more important than others
• Want to be sure you understand basics of CS
Exam Coverage

The high-level view:
1. Functional programming
2. Computer organization
3. Imperative and object-oriented programming
4. Theory of computation
Course Eval Break!
Functional Programming

Be sure you know (and command):

- List indexing and slicing (including ::-1)
- Recursion (multiple base cases)
- Functions that return functions
- List comprehensions:
  - \([x**2 \text{ for } x \text{ in } L \text{ if } x < 100]\)
- map, reduce, and filter
- lambda

Especially lambda!
Write \texttt{compose(L)}, where \( L \) is a list of functions. Compose returns a single \textit{function}, \( f(x) \), that calculates \( L[0] (L[1](L[2]...(x)) \)

Example:

\[
f = \text{compose}([\log, \sin])
\]

\[
f(\pi/2) == 0.0
\]

\[
f(\pi/4) == -0.34657...
\]
Functional Challenge

def compose(L):

def compose(L):
    if L == []:
        return lambda x: x
    return lambda x: \
        L[0](compose(L[1:])(x))
Computer Organization

Important things:

• Binary numbers (can you count in binary?)
• Boolean equations
• Minterm expansion principle
• Hmmm programming
  • Be sure you understand function calls and how r13, r14, and r15 are used
Minterm Expansion

Be systematic!

0 0 0 1
0 0 1 1
0 1 0 0
0 1 1 1
1 0 0 0
1 0 1 1
1 1 0 0
1 1 1 1
A Hmmm Problem

What does this program do?

00 read r1
01 setn r2 0
02 jltzn r1 06
03 add r2 r2 r1
04 addn r1 -1
05 jgtzn r1 03
06 write r2
07 halt

Is there anything ADDitional needed here?

Oooh, that’s awful!
Imperative Programming

What to know:

- Nested for loops
- `for i in range(len(L))` vs. `for i in L`
- try/except
- Deep and shallow copy

What NOT to know:

- Huffman coding
Consider a matrix (list of lists) $\mathbf{M}$, containing strings or integers. For elements not on the diagonal, if they are numbers, convert them into their (floating-point) reciprocals. All other elements stay the same.

Example: $\begin{bmatrix} [1, 2, "spam"] & \mathbf{[1, 0.5, "spam"]}, \\ [3, "hi", 5], & \Rightarrow [0.33, "hi", 0.2], \\ [7, 4, 9] \end{bmatrix}$ $\Rightarrow [0.14, 0.25, 9]$
for i in
  for j in
for i in range(len(M)):
    for j in range(len(M[0])):
        if i != j:
            try:
                M[i][j] = 1.0 / M[i][j]
            except:
                pass
Object-Oriented Programming

What to know:

• `__init__`, `__str__`, `__repr__`
• `self`
• `__add__`, `__mul__`, etc.
• Methods, getters, setters
• Inheritance
Object-Oriented Exercise

Implement the `__add__` function for a Rational class. The numerator is in `self.n`, the denominator in `self.d`. Assume `self.gcd(a, b)` will return the greatest common divisor of `a` and `b`.
class Rational:
    # ...
    def __add__(self, other):
        pass
class Rational:
    # ...
    def __add__(self, other):
        n = self.n * other.d \ + other.n * self.d
d = self.d * other.d
gcd = self.gcd(n, d)
return Rational(n / gcd, d / gcd)
Theory of Computation

What to know:
• Finite state machines
• FSMs can’t count
• Existence of uncomputable functions
• Uncomputability of halting problem
• Turing machines

What \textit{NOT} to know:
• How to prove uncomputability
FSM Exercise

Draw an FSM that accepts strings in which each 1 is followed by two or more 0’s.
Extra credit: draw an FSM that accepts strings in which each 1 is followed by two or more 0’s, plus strings in which each 0 is followed by two or more 1’s.
FSM Exercise (1)
FSM Exercise (1)
FSM Exercise (2)
FSM Exercise (2)