



CSCI 120

Introduction to CompSci and Programming I

Midterm Review



Jetic Gū

Overview

- Focus: Python Programming
- Architecture: von Neumann
- Core Ideas:
 1. Review some Data Structures

Basic Things about Python

What Python?

- Named after the comedy group monty python
- Interpretive programming language
- Two ways of using python
 - Interactive mode / Python console
 - Script

Built-in Functions

- `input(...)`
- `print(...)`
- `max(...)`, `sum(...)`, `min(...)`

Immutable Simple Data Types

- int
- float
- string
- bool

Data Structures

Lists, Strings, Tuple, Dictionary, etc.

Data Structures

- Data structures are particular ways of storing data to make some operation easier or more efficient. That is, they are tuned for certain tasks
- Data structures are suited to solving certain problems, and they are often associated with algorithms.

Kinds of data structures

Roughly two kinds of data structures:

- built-in data structures, data structures that are so common as to be provided by default
- user-defined data structures (classes in object oriented programming) that are designed for a particular task

Python built in data structures

- Python comes with a general set of built in data structures:
 - int
 - float
 - lists
 - tuples
 - string
 - dictionaries
 - others...

Lists

- An ordered group of items
- Does not need to be the same type
- List notation

```
A = [1, "This is a list", c]
```

Methods of Lists

- `alist.append(x)`
 - adds an item to the end of the list
- `alist.extend(L)`
 - Extend the list by appending all in the given list L
- `alist.insert(I, x)`
 - Inserts an item at index I
- `alist.remove(x)`
 - Removes the first item from the list whose value is x

Examples of other methods

- Define and print List, then count values

```
a = [66.25, 333, 333, 1, 1234.5]
print (a.count(333), a.count(66.25), a.count('x')) //calls method
2 1 0          //output
```
- Return the first index where the given value appears

```
a.index(333)
1          //output
```
- Reversing a list **(in place)**

```
a.reverse() //Reverses order of list
[333, 1234.5, 1, 333, -1, 66.25]
```
- Sorting **(in place)**

```
a.sort()
[-1, 1, 66.25, 333, 333, 1234.5]
```

The del statement

- Deleting at index: `del a[0]`
- Deleting at index range: `del a[2:5]`

Tuples

- Tuple
 - A number of values separated by commas
 - Immutable
 - Cannot assign values to individual items of a tuple
 - However tuples can contain mutable objects such as lists
 - Single items must be defined using a comma
 - `Singleton = 'hello',`

Dictionaries

- Indexed by **keys**
 - This can be any immutable type (strings, numbers...)
 - Tuples can be used if they contain only immutable objects

Looping Techniques

- `adict.items()`
- for retrieving key and values through a dictionary

```
stuff = {  
    'name': 'Jetic',  
    'age': '44',  
    'favourite food': 'cheese'}  
for k, v in stuff.items():  
    print("His " + k + " is " + v)
```

Looping Techniques

- `zip()`
- for looping over two or more sequences

```
key = ['name', 'age', 'favourite food']
val = ['Jetic', '44', 'cheese']
for k, v in zip(key, val):
    print("His " + k + " is " + v)
```

```
// Result
```

```
His name is Jetic
```

```
His age is 44
```

```
His favourite food is cheese
```

Comparisons

- Operators “in” and “not in” can be used to see if an item exists in a sequence
- Comparisons can be chained
 - $a < b == c$
This tests whether a is less than b and that b equals c