

CS:1210 – Summer 2019 Final Exam Topics

Numeric Types

- `int`, `float`, `bool`, `complex`
- Converting between types

Strings

- Creating a string
 - Literal (single, double, and triple quotes)
 - Repetition
 - With constructor
- Special characters: `\n` `\t` `\b` `\r` `\\` `\'` `\"`
- White space
- Indexing
- Slicing
- Concatenation
- Comparing strings
- `in` operator
- Formatting strings
- Iterating over a string
- String methods:
 - `center`, `count`, `endswith`, `find`, `format`, `index`, `isalnum`, `isalpha`, `isdigit`, `islower`, `isnumeric`, `isupper`, `join`, `ljust`, `lower`, `lstrip`, `replace`, `rfind`, `rindex`, `rjust`, `split`, `startswith`, `strip`, `swapcase`, `title`, `upper`

Lists

- Creating a list
 - Literal (providing each element explicitly)
 - Repetition
 - Comprehension
 - With constructor
- Indexing
- Slicing
- Concatenation
- Comparing lists
- `in` operator
- Iterating over a list

- Deep copy vs. shallow copy
- List methods:
 - **append, clear, count, extend, index, insert, pop, remove, reverse, sort**

Tuples

- Creating a tuple
 - Literal (providing each element explicitly)
 - Repetition
 - Comprehension
 - With constructor
- Indexing
- Slicing
- Concatenation
- Comparing tuples
- **in** operator for tuples
- Iterating over a tuple
- Tuple methods:
 - **count, index**

Dictionaries

- Creating a dictionary
 - Literal (providing each key, value pair explicitly)
 - Comprehension
 - Using output of **zip**
 - With **fromkeys** method
 - With Constructor
- Accessing values
- Iterating over a dictionary
- Comparing dictionaries
- **in** operator
- Dictionary methods:
 - **clear, fromkeys, get, items, keys, pop, popitem, values**

Files and Web Pages

- Opening, reading, and closing text files and web pages
- Current working directory
- Module **os**

Operators

- `+` `-` `*` `/` `//` `%` `**`
- Assignment version of above operators
- unary `+` `-`
- `=`
- Multiple assignment, iteration with multiple assignment
- `<` `<=` `==` `>=` `>` `!=` `is`
- Chained relational operators
- `and` `or` `not`
- `in`
- Order of precedence

Built-In Functions

- `abs`, `all`, `any`, `chr`, `dir`, `filter`, `id`, `input`, `isinstance`, `len`, `map`, `max`, `min`, `open`, `ord`, `pow`, `print`, `range`, `reversed`, `round`, `slice`, `sorted` (with `key=`), `sum`, `super`, `type`, `zip`

User Defined Functions

- Defining a function with `def`
- Calling a function
- Parameters, default values, arguments, named arguments
- `return` statement
- Flow control with functions
- Namespace and scope
- Lambda expressions

Control Structures

- Sequence, selection, repetition
- `if`, `elif`, `else`
- `for`, `while`
- nested loops

Modules

- `import` statement
- `from ... import *`
- `copy`, `math`, `os`, `random`, `string`, `sys`, `turtle`

Recursion

- When to use recursion
 - Finding the smaller instance of a problem
 - Using the smaller instance
 - Identifying the base case
- Writing a recursive solution
- The call stack
- Efficiency of recursion

Classes

- Defining a class with **class**
- **self**
- Code reuse, encapsulation, information hiding, public interface
- Inheritance
 - Superclass, subclass
- Class vs. object
- Class variable, instance variable, local variable
- Programmer defined methods
 - Accessor, mutator, predicate
- Special methods:
 - `__init__`, `__str__`, `__repr__`
 - `__pos__`, `__neg__`
 - `__add__`, `__sub__`, `__mul__`, `__div__`, `__floordiv__`,
`__mod__`, `__pow__`
 - `__radd__`, `__rsub__`, `__rmul__`, `__rdiv__`, `__rfloordiv__`,
`__rmod__`, `__rpow__`
 - `__lt__`, `__le__`, `__eq__`, `__ge__`, `__gt__`, `__ne__`
 - `__int__`, `__float__`, `__complex__`, `__abs__`, `__len__`
 - `__getitem__`

Miscellaneous

- Command line arguments
- Use of indentation
- Continuation with `\`
- Comments
- **pass** statement