# CSc 110 Final
# Cheat Sheet

```
def name(parameters):
    statement(s)
    ...
    return expression
```

```
for name in range(start, stop + 1):
    statement
    statement
    ...
    statement
```

```
while(condition):
    statements(s)
```

```
variable = type(input(prompt))
```

```
if (test):
    statement(s)
elif (test):
    statement(s)
else:
    statement(s)
```

## `Math` class

| Function name | Description |
|---|---|
| `ceil(`*value*`)` | rounds up |
| `floor(`*value*`)` | rounds down |
| `log(`*value, base*`)` | logarithm |
| `sqrt(`*value*`)` | square root |
| `sinh(`*value*`)`<br>`cosh(`*value*`)`<br>`tanh(`*value*`)` | sine/cosine/tangent of an angle in radians |
| `degrees(`*value*`)`<br>`radians(`*value*`)` | convert degrees to radians and back |

| Constant | Description |
|---|---|
| `e` | 2.7182818... |
| `pi` | 3.1415926... |

### Other math functions:

| Function name | Description |
|---|---|
| `abs(`*value*`)` | absolute value |
| `min(`*value1, value2*`)` | smaller of two values |
| `max(`*value1, value2*`)` | larger of two values |
| `round(`*value*`)` | nearest whole number |

## `String` functions

| Function | Description |
|---|---|
| `find(`***str***`)` | index where the start of the given string appears in this string (-1 if not found) |
| `substring(`***index1, index2***`)`<br>or<br>`substring(`***index1***`)` | the characters in this string from index1 (inclusive) to index2 (exclusive);<br>if index2 is omitted, grabs till end of string |
| `lower()` | a new string with all lowercase letters |
| `upper()` | a new string with all uppercase letters |

`len(`**thing**`)` – returns the length of whatever (string, list, etc) is passed to it

### Declaring and using lists

**name** = [**value**] * **length**
**name**[**index**] = **value**

### Random

`randint(`**min**, **max**`)`

## Sets

**name** = set()
**name**.add(**value**)

## Dictionaries

**name** = {}
**name**[**key**] = **value**
**othername** = **name**[**key**]

## Classes

**Field** (data inside each object)
self.**name** = **value**

**Method** (behavior inside each object)
def **name**(self, **parameters**):
    **statements**

**Constructor** (code to initialize new objects)
def __init__(self, **parameters**):
    **statements**

**__str__ method** (called when an object is printed)
def __str__(self):
    code that produces and returns a string

### Inheritance

class **name**(**superclass**):

### Critter classes

```
class name(Critter):
    def __init__(self):
        fields
    def eat(self):
        statement(s) that return True (eat) or False (don't eat)
    def fight(self, opponent):
        returns either ATTACK_ROAR, ATTACK_POUNCE, or ATTACK_SCRATCH
    def get_color(self):
        statement(s) that return a color
    def get_move(self):
        statement(s) that return either DIRECTION_NORTH, DIRECTION_.SOUTH,
        DIRECTION_EAST, DIRECTION_WEST, or DIRECTION_CENTER
    def __str__(self):
        statement(s) that return a string
```