Winter 2014

Computer Science 1MD3
Introduction to Programming

Michael Liut (liutm@mcmaster.ca)
Brandon Da Silva (dasilvbc@mcmaster.ca)
Ming Quan Fu (fumq@mcmaster.ca)

www.michaelliut.ca/cs1md3

Counters

- Used in loops to "count" how many times you are iterating through the loop
  - If counters weren't used, the loops would go on forever
- Typically used in the while-loop condition
- Another use of a counter is to keep a count during a loop of a specific condition that is met and keeping track of how many times that condition occurs (eg. assgn1_3.py)

“Counters” Example

“Counter” Example

```
i = 1
while (i <= 3):
    print (str(i))
    i++              # Increments the counter

print (“\nThe loop was executed “ + i + “ times.”)
```

```
1
2
3

The loop was executed 3 times.
>>
```

Lists

Working with Lists in Python

grades = [89, 76, 92, 83]	
grades.append(94) print(grades) >>[89, 76, 92, 83, 94]	Appending a number to a list adds it to the end of the list. You can even include expressions in the parentheses!
grades.insert(2, 67) print(grades) >>[89, 76, 67, 92, 83]	The first number in the brackets when inserting a number tells Python where in the list you want to put it. The list starts at position 0.
print(grades[3]) >> 83	Like in Java, Python can call certain variables from the list

Working with Lists - 2

grades = [89, 76, 92, 83]	
grades + [19, 23, 34] >>[89, 76, 92, 83, 19, 23, 34]	Concatenation works similar to appending to the end of a list but it can be more than one element
grades[1:3] >>[76, 92, 83]	Using "slices" can retrieve a range of elements from a list.
len(grades) >>4	The len() function retrieves the size of the list (i.e. how many elements it contains)

Lists - Example

Iterating over a list

```
grades = [89, 76, 92, 83]

for i in range(0, len(grades)):
    print (grades[i])
```

```
89
76
92
83
>>
```

Note: the range of values goes from 0 to 4 (i.e. len(grades)). What the range function is actually doing is counting from its starting value (0) to it's end value (4) but it actually does not include the 4. So it is going from 0 to 3 but it is still iterating over 4 values.

File Reading & Writing

www.michaelliut.ca
/cs1md3

File Writing

- MAKE SURE YOUR FILE IS IN THE SAME DIRECTORY AS YOUR SOURCE CODE FILE
- In Python, the open() method returns a file object.
- open(<file name>, <mode>)
  - Mode refers to what you would like to do with the file (read, write, etc…).
  - We are setting the mode to "w" which stand for writing.
- Example:

File Writing

```
fileName = open("file.txt", "w")
fileName.write("Hello world! ")
fileName.close()
```

Appending Text to Files

- Appending text to files means that you are adding text to an already existing file, rather than deleting the file and adding the text to a new file.
- Very similar to the coding for text writing, the only difference is changing the "mode" from "w" to "a" (in many other languages it is as simple as changing the Boolean value from false to true).
- Example:

File Appending

```
fileName = open("file.txt", "a")
fileName.write("My name is Billy!")
fileName.close()
```

File Reading

- File reading is very similar to file writing and appending, the mode is set to "r" for reading.
- As well as using the read() or readLine() method instead of write()
  - read() reads everything in the file
  - readLine() reads a single line from the file
- Example:

File Reading

```
fileName = open("test.txt", "r")
fileText = fileName.readLine()
print(fileText)
fileName.close()
```

Functions

www.michaelliut.ca
/cs1md3

Making Functions

- Functions are based purely on indentation (white-case sensitive)
- There is no need to declare what types the variables are
- Function blocks begin with the keyword **def**, which stands for define, followed by the name of the function and then ( )
- Function must return a value, otherwise 'None' is returned
- Example:

Python

```
def bmi (weight, height):
        bmi = weight/(height*height)
        return bmi
```

Using the Function

- Using the function you created
- Example:

Python

```
print(bmi(75, 1.5))

>>33.3333333333
```

Classes

www.michaelliut.ca
/cs1md3

Creating Classes

Creating a Class

```
class Triangle:
    def _init_(self, base= 0, height = 0):
        self.base
        self.height

    def setBase(self, base):
        self.base = base

    def setHeight(self, height):
        self.height = height

    def findArea(self):
        area = (self.base*self.height)/2
        return (area)
```

Main Function

Main Function

```
def main():
        tri = Triangle()
        tri.setBase(4)
        tri.setHeight(2)
        triArea = tri.findArea
        print(triArea)

>> 2
```

Assignment 2

www.michaelliut.ca
/cs1md3

Assignments Due

Assignment 1 Due:
THURSDAY JANUARY 30, 2014
BY 11PM

Assignment 2 Due:
WEDNESDAY FEBRUARY 26, 2014
BY 11PM

The End

www.michaelliut.ca/cs1md3