
CMSC 201 Fall 2016

Midterm – Vocabulary and Practice Problems

Vocabulary: The following are terms that you should be familiar with, and that will likely show up on the midterm. We won't be asking you to "define" any of these terms, but you should know what they each mean and, if they are a technique or programming concept, how to use them. This list is not guaranteed to be exhaustive, but it covers most of the concepts you'll need.

Algorithm	List
Binary	Logic
Boolean	Loop
Branching	Magic Number
Bug	Main
Case Sensitive	Modularity
Casting	Modulus (or Modulo/Mod)
Code	Mutable (and Immutable)
Comment	Nested (e.g., loops, conditionals)
Concatenation	Operator (e.g., assignment, comparison, logical)
Conditional	Order of Operations
Constant	Parameter (e.g., formal, actual)
Control Structure	Priming Read
Debugging	Program
Decision Structure	Pseudocode
Error (e.g., logic, syntax)	Return
Expression	Scope
Function	Selection
Incremental Development	Sentinel Loops
Index	Sequential
Indexing	Slicing
Infinite Loop	String
Input and Output	Syntax
Integer	Truth Table
Integer Division	Value
Iteration	Variable
Keyword	

Practice Problems: The following are all of the practice problems from the slides that we have used so far this semester. You are also encouraged to take another look at the livecoding examples from Lectures 6, 7, 8, and 9. (Solutions are available on the course website at the bottom of the “Documents” page.)

There are also two files available for you to complete as practice. You can find them at the following locations. Use the “cp” command to copy them into your own directory so you can edit them.

```
/afs/umbc.edu/users/k/k/k38/pub/cs201/stringPractice.py  
/afs/umbc.edu/users/k/k/k38/pub/cs201/listPractice.py
```

LECTURE 01 – INTRODUCTION

What are the decimals equivalents of...

101
1111
100000
101010
1000 0000

What are the binary equivalents of...

9
27
68
100

LECTURE 02 – ALGORITHMIC THINKING

Create a program to calculate the weekly pay of an hourly employee.

Create a program to answer the question “Is a number positive?”

Write an algorithm that counts from 1 to 20.

Find the syntax errors in each line of code below:

```
prnit("Hello")  
print("What"s up?)  
print("Aloha!)  
print("Good Monring")
```

LECTURE 03 – VARIABLES

What will each of the following do?

```
print("Hello')
Print('Hello')
print('Hello World')
```

Are the following variable names legal or illegal in Python?

```
1spam
raise1
Spam_And_Eggs
```

What will the following code snippet print?

```
a = 10
b = a
a = 3
print(b)
```

Write, on paper or on your computer, a program that asks the user for two numbers and prints out the average.

Pretend you're writing a program to compute someone's weight grade. You have so far:

```
hwWeight          = 0.4
examWeight        = 0.5
discussionWeight  = 0.1
```

Write a program that then asks the user for their homework grade, exam grade, and discussion grade and prints out their total grade in the class.

LECTURE 04 – EXPRESSIONS

Which of the following examples are correct?

```
500 = numStudents
numStudents = 500
numCookies * cookiePrice = total
mpg = miles_driven / gallons_used
"Hello World!" = message
_CMSC201_doge_ = "Very learning"
60 * hours = days * 24 * 60
```

Complete the following examples that use different arithmetic operators:

```
7 / 5 =
7 // 5 =
2 / 8 =
2 // 8 =
4 // 17 // 5 =
```

```
7 % 5 =
5 % 9 =
17 % 6 =
22 % 4 =
48692451673 % 2 =
```

```
4 / 3 =
4 // 3 =
4 // 3.0 =
8 / 3 =
8 / 2 =
5 / 7 =
5 // 7 =
```

LECTURE 05 – DECISION STRUCTURES

What are the following comparisons asking?

`a >= b`

`a == b`

Evaluate the following code snippets – what do they print out?

`a = 10`

`b = 20`

`c = 30`

`bool1 = int(a == a)`

`bool2 = a >= 10`

`bool3 = (a == a) + (b == b) + (c == c)`

`print(bool1, bool2, bool3)`

`a = 10`

`b = 20`

`c = 30`

`ex1 = a < b`

`ex2 = a < b and b < c`

`ex3 = a + b == c and b - 10 == a and c / 3 == a`

`print (ex1, ex2, ex3)`

`a = 4`

`b = 5`

`c = 6`

`d = True`

`e = False`

`bool1 = d and (a > b)`

`bool2 = (not d) or (b != c)`

`bool3 = (d and (not e)) or (a > b)`

`bool4 = (a % b == 2) and ((not d) or e)`

`bool5 = (d + d) >= 2 and (not e)`

`bool6 = (not e) and (6*d == 12/2)`

`bool7 = (d or (e)) and (a > b)`

Create a program that converts from Celsius to Fahrenheit, and prints a warning when the weather is extreme. Over 90 degrees Fahrenheit will cause a hot weather warning; lower than 30 degrees Fahrenheit will cause a cold weather warning.

Complete these truth tables

bool1 = a and b

Value of a	Value of b	Value of bool1
True	True	
True	False	
False	True	
False	False	

bool2 = a or b

Value of a	Value of b	Value of bool2
True	True	
True	False	
False	True	
False	False	

bool3 = not a

Value of a	Value of bool3
True	
False	

bool4 = a and (b or c)

Value of a	Value of b	Value of c	Value of bool4
True	True	True	
True	True	False	
True	False	True	
True	False	False	
False	True	True	
False	True	False	
False	False	True	
False	False	False	

LECTURE 06 – STRINGS (AND DECISION CONTINUED)

You have just been flown to an island where there are a wide variety of dinosaurs. You are unsure which are dangerous so come up with some rules to figure out which are dangerous and which are not.

A computer science professor gives a five-point quiz at the beginning of every class. Possible grades are as follows:

5 points: A 3 points: C 1 point: F
4 points: B 2 points: D 0 points: F

To print out the letter grade based on the raw points, what would the code need to look like?

You recently took a part-time job to help pay for your student loans at a local cell phone store. If you sell at least \$1000 worth of phones in a pay period, you get a bonus. Your bonus is 3% if you sold at least 3 iPhones, otherwise your bonus is only 2%.

Complete the repetition and concatenation examples below.

```
animal = "dogs"
```

```
animal*3
```

```
animal*8
```

```
"spam" + "eggs"
```

```
"Spam" + "And" + "Eggs"
```

```
3 * "spam"
```

```
"spam" * 5
```

```
(3 * "spam") + ("eggs" * 5)
```

Complete the slicing and indexing examples below.

```
greet = "Hello Bob"
```

```
greet[0]
print(greet[0], greet[2], greet[4])
x = 8
print(greet[x - 2])
greet[-1]
greet[-3]
greet[0:2]
greet[5:9]
greet[:5]
greet[1:]
greet[:]
greet[2:-3]
greet[-6:-2]
greet[-6:6]
greet[-9:8]
```

Create a username from the user's first and last name. The rules for a username are the first initial, followed by the first 7 letters of the last name, all in lowercase.

For example:

Donna Rostenkowski becomes **drostenk**

Barack Obama becomes **bobama**

LECTURE 07 – WHILE LOOPS

Use a `while` loop to count from 1 up to and including 20.

Fix the following infinite loops, or state why they won't work.

```
age = 0
while age < 18:    # can't vote until 18
    print("You can't vote at age", age)
print("Now you can vote! Yay!")
```

```
while True:
    # ask user for name
    name = input("What is your name? ")
print("Hello", name + "!")
```

```
cookiesLeft = 50
while cookiesLeft > 0:
    # eat a cookie
    cookiesLeft = cookiesLeft + 1
print("No more cookies!")
```

```
grade = ""
name = ""
while name != "Hrabowski":
    # get the user's grade
    grade = input("What is your grade? ")
print("You passed!")
```

Complete the while loop below so it asks for the user's grade until they report that they earned an "A" or a "B" as their grade.

```
grade = ""
while _____:
    # get the user's grade
    grade = input("What is your grade? ")
print("You passed!")
```

Write a program that allows the user to try guessing a password. It should allow them to guess the password up to three times.

LECTURE 08 – LISTS

Create a grocery list program that allows the grocery list to be as long as the user wants, using a while loop and a sentinel value of “STOP”
Print out the grocery list (item by item) at the end. Use only while loops.

Which of the following works with lists? What effects does it have?

- Concatenation (+)
- Repetition (*)
- Indexing
- Slicing
- `.lower()` and `.upper()`
- `len()`

What are the indexes for numbers 3, 4, 8, and 9 in the two-dimensional list to the right?

1	2	3
4	5	6
7	8	9

Update the `names` list in the following ways.

- Add a “b” and a “y” to the end of “Bob”
- Print out the second letter in “Evan”
- Change “Alice” to “Alyce”
- Add “Ahmed” to the end of the list of names
- Add “Eve” to the front of the list of names
- Change “Evan” to “Kevin”

A	l	i	c	e
B	o	b		
E	v	a	n	

`names`

LECTURE 09 – FOR LOOPS

Evaluate the following code.

```
animals = ['cat', 'dog', 'eagle', 'ferret',
          'horse', 'lizard']
print("The best animal is a", animals[3])
animals[5] = "mouse"
print("The little animal is a", animals[5])
print("Can a", animals[4], "soar in the sky?")
```

Write code that uses a `for` loop to find the average from a list of numbers.

What does the following code do?

```
music = "jazz"
for c in music:
    print(c)
```

What does the following code do?

```
myList = [1, 2, 3, 4]
for listItem in myList:
    listItem = 4
print("List is now:", myList)
```

Given a list of strings called `food`, use a `for` loop to print out that each food is yummy!

What does the following code do?

```

for i in range(0, 20):
    print(i + 1)

for p in range(4):
    print(p)

for a in range(5,10):
    print(a)

for b in range(10,5):
    print(b)

for c in range(-10, -5):
    print(c)

threeA = list(range(2, 11, 2))
print(threeA)
threeB = list(range(3, 28, 5))
print(threeB)
downA = list(range(10, 0, -1))
print(downA)

print("Counting by fives...")
for num in range(5, 26, 5):
    print(num)

prac1 = list(range(50))
print(prac1)
prac2 = list(range(-5, 5))
print(prac2)
prac3 = list(range(1, 12, 2))
print(prac3)

```

You are running a kennel with space for 5 dogs.

You ask your 3 assistants to do the following, using the list of dogs in your office:

1. Tell you all of the dogs in the kennel
2. Tell you what pen number each dog is in
3. Later, all the dogs have been picked up, and someone dropped off their 5 German Shepherds, so the list in your office needs to be updated

Write code that will print out, for the numbers 1 through 20, whether that number is even or odd

Sample output:

```
The number 1 is odd
The number 2 is even
The number 3 is odd
```

Using the list names, code the following:

names =

Alice	Bob	Eve
-------	-----	-----

Print "Bob sends a message to Alice"

Change the first element of the list to Ann

Print "BobBobAnnEve"

Write a program that asks the user for input:
The height and width of a parallelogram
Print a parallelogram with those dimensions to the user's screen

Sample output:

```

What is the height of your parallelogram? 4
What is the length of your parallelogram? 7
*****
*****
*****
*****
*****

```

Write a program that asks the user to enter a positive integer, and calculates the sum of $1^2 + 2^2 + \dots + n^2$ and prints it to the screen

For example, if the user entered 5, the program would calculate

$$1^2 + 2^2 + 3^2 + 4^2 + 5^2$$

$$= 1 + 4 + 9 + 16 + 25 = 55$$

Sample output:

```

Enter a positive integer: 4
The sum of the first 4 squares is 30.

```

LECTURE 10 – FUNCTIONS

What do the following code snippets output?

```
for i in range(5):  
    print(i)
```

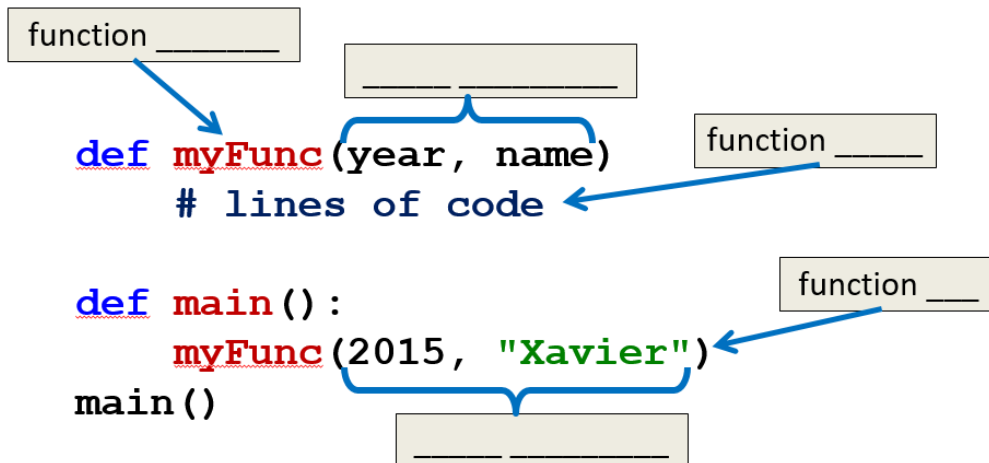
```
for i in range(-3, -13, -3):  
    print(i)
```

Write a function called `avg()` that takes in a list of numbers, calculates the average, and prints the result to the screen

Write a function that prints out the lyrics to a song – you can pick any song you like! If it has a chorus, put that in a separate function, and call it when necessary.

LECTURE 11 – FUNCTIONS (CONTINUED)

Label the following parts of a function:



What happens for each of the following snippets of code? What would you do to fix each error?

```

def test():
    print("In the fxn")
    var = 3
var2 = test()
print(var2)

```

```

def test():
    print("In the fxn")
    return 3
var2 = 7
test()
print(var2)

```

```

def test():
    print("In the fxn")
    return 3
var1, var2 = test()

```

What is wrong with the following code? Update the code so that the problem is fixed.

```
def main():
    amount = 1000
    rate = 0.05
    addInterest(amount, rate)
    print(amount)

def addInterest(balance, rate):
    newBalance = balance * (1 + rate)
    balance = newBalance
main()
```

Update the following code so that it works with a list of balances.

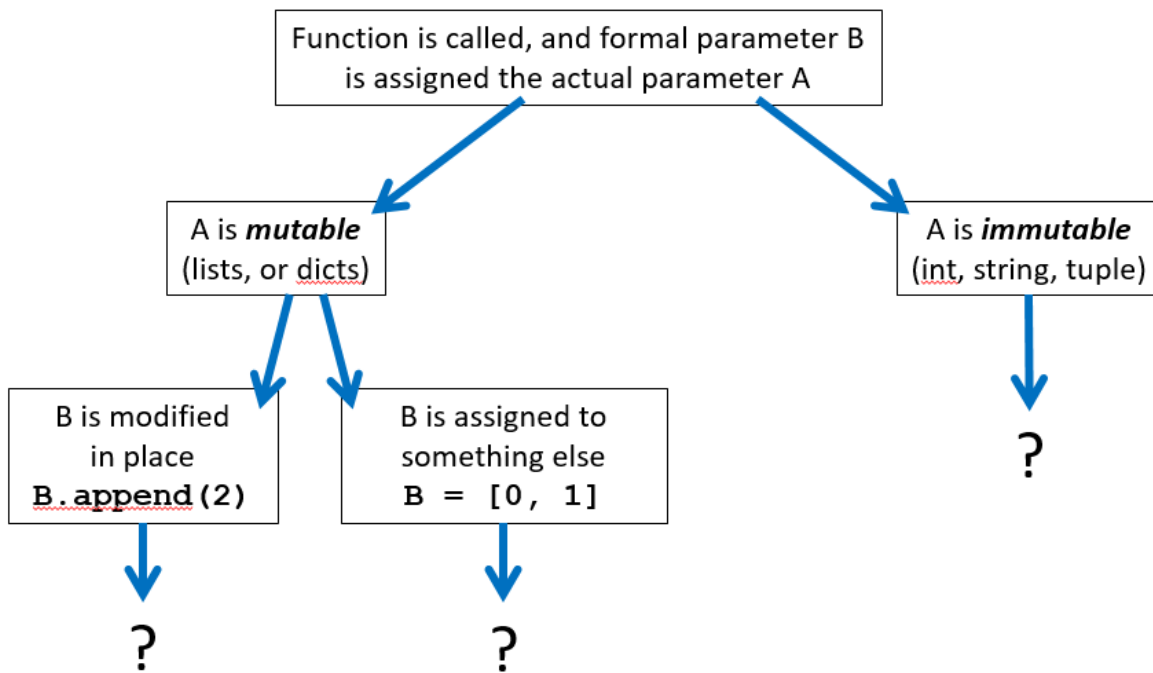
```
def addInterest(balances, rate):
    # add your code here

def main():
    amounts = [1000, 2200, 800, 360]
    rate = 0.05
    addInterest(amounts, rate)
    print(amounts)

main()
```

LECTURE 12 – PROGRAM DESIGN AND MODULARITY

Complete the following chart. Will A change if B changes in each of these examples?



What does the following code snippet do?

```

list1 = ["red", "blue"]
list2 = list1[:]
list2[1] = "yellow"
list3 = list(list1)
list3.append("purple")
print("original:      ", list1)
print("deep copy1:     ", list2)
print("deep copy2:     ", list3)
  
```

What does the following code snippet do?

```
list1 = ["red", "blue"]
list2 = list1
list2.append("green")
list2[1] = "yellow"
print("list1 (end): ", list1)
print("list2 (end): ", list2)
```

Congratulations, you made it to the end!