CMSC 201 Midterm 2 Review Worksheet

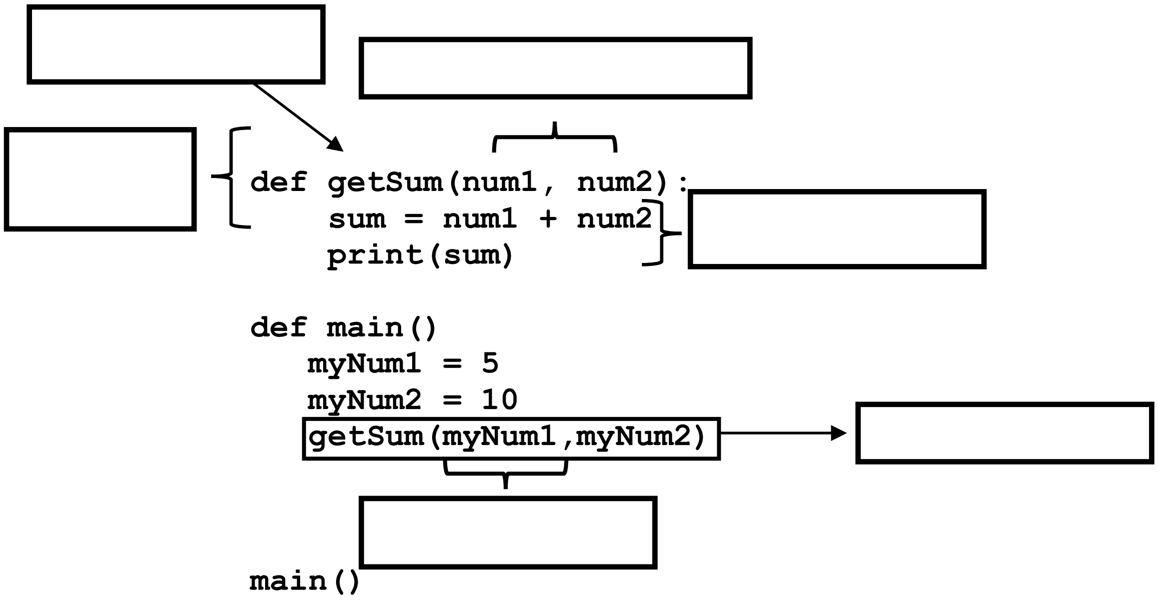
This worksheet is ***NOT*** guaranteed to cover every topic you might see on the exam. It is provided to you as a courtesy for additional practice to help you study. You should also be reviewing the course notes and assignments as part of your preparation for the exam. Answers will be provided a day or two before the first exam.

*** ***DO NOT WAIT UNTIL ANSWERS ARE RELEASED TO START*** ***

You are encouraged to work with other students in the class to confirm your answers and solidify your understanding of the material. You are also encouraged to seek help from TAs during office hours should you be stuck on question.

---

1. Name the parts of the function:



```
def getSum(num1, num2):
    sum = num1 + num2
    print(sum)

def main()
   myNum1 = 5
   myNum2 = 10
   getSum(myNum1,myNum2)

main()
```

2. In the program used in #1, what are the scopes of the following variables:

    - **sum**
    - **num1**
    - **num2**
    - **myNum1**
    - **myNum2**

3. What are two advantages of using functions?
4. What are some examples of function in python that return values?
5. What happens when Python encounters **`return`**?
6. Describe the difference between Top-down and Bottom-Up Design
7. How would you access the 3rd element in the 4th row of **`gameBoard`**, a 2D list?
8. Why is a list of strings like a two-dimensional list?
9. What is the difference between mutable and immutable structures? Name a few examples of each.
10. What is the difference between a shallow copy and a deep copy of a list?
11. Describe what the following line does: **`myNums = list(range(10,2,-2))`**
12. What is one benefit of using a for-loop over a while-loop?
13. What are some benefits of incremental programming?
14. Python uses a ___________ to keep track of function calls.
15. Describe a LIFO data structure
16. A recursive function must have at least one __________ __________, which is when the function ends or a result is returned, and it must also have at least one ______________ _____________, which is when the function is called again with new inputs.
17. Describe what short-circuiting is and give an example of an if-statement that will short-circuit.
18. String Formatting: write down the output of each of these snippets of code:

    - **`print(“hello {:*^9}”.format(“world”))`**
    - **`print(“I made ${:.2f}”.format(132.4920))`**
    - **`print(“{:05d}”.format(201))`**

19. Write a for-loop to perform each of the following actions:
    - Print out every item in a 2D list
    - Print out the first letter in the first item in a list, 2nd letter in the 2nd item in the list, 3rd letter in the 3rd item, etc
    - Create a 4x4 2D list that counts up from 1-16

20. Write each of these functions and correctly use them in main():
    - **`getListSum(numbers)`** – given a list of numbers, returns the sum of the list
    - **`decryptCode(code, key)`** – given a code and a key, print out the "Decrypted" string
        i. example: code = "dsaggrwofdsoedadfdqljtsujs8c048k", key = 4
            1. should print "goodluck"
    - **`factors(number)`** – given an integer, return a list of ints of the factors of that number
        - ex: if number = 10, [1,2,5,10] should be returned
    - **`pyramid(character, height)`** – given a character, return a 2D list that contains a pyramid of size **`height`**
    - **`getReverse(word)`**– given a string and using recursion, return the reverse of the string

21. Code Evaluation: What is the output of the following code segments?

```
def createSquareList(dimension):


  start = dimension * dimension
  board = []


  for i in range(dimension):
    row = []
    for j in range(dimension):
            row.append(start)
            start -= 1
        board.append(row)

    return board

def main():
    print(createSquareList(3))

main()
```

```
def calcExponent(base, power):

    if power == 1:
        return base
    else:
        return (base * calcExponent(base, power - 1))

def main():
    answer = calcExponent(2,3)
    print(answer)
main()
```

```
def myFunction(line):
    if len(line) == 0:
        return ""
    else:
        return line + "\n" + myFunction(line[1:])
```

```
def main():
    print(myFunction("watermelon"))
main()
```

```
def main():


    list1 = ['h','e','l','l','o']
    list2 = list1
    list3 = list2[:]
    list2.append("!")
    list3.append("?")
    print(list1)
    print(list2)
    print(list3)


main()
```

22. Find and fix the errors (syntax or logic) in the following code snippets:

```
# this function returns a string containing every other letter in
# word
def everyOther(word):


    final  = ""
    for i in range(word):
        if i % 2 == 0:
            final = word[i]


    return final


def main():
    everyOther("grapefruit")
    print(final)
main()
```

```
# this function creates a 2D list using the height, width, and symbol. It does
# not contain any shallow copies of rows
def createList(height, width, symbol):

    board = []

    for i in range(len(height)):
        row = []

        for j in range(width):
            row.append(symbol)

        board.append(row)

def main():

    # creates 5 rows of 5 @s
    myList = createList(5,"@")

    print(board)

main()
```