**CS110 Computer Science Fundamentals Final Exam 27th April 2023**

*Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Please observe the Emory College Honor Code while taking this test. NOTE: You* ***must*** *show all work; the final answer alone will give you little credit, if any.*

1. (5 points) Some arbitrary byte in a computer’s memory contains the bit pattern 01010111 This must represent

(a) the code for the ASCII character ‘W’

(b) the integer number 87

(c) the floating point number corresponding to PI i.e. 3.14

(d) one entry of an example BankAccount object

(e) impossible to determine without additional context

2. (5 points) What is the binary representation, using 8 bits, of the decimal integer 89

3. (5 points) What are the three core constructs that determine control flow in any computer program whatsoever?

(a) logic gates, printer, and mouse

(b) touchscreen, keyboard, memory

(c) sequential execution, decision, repetition

(d) objects, dictionaries, lists

(e) addition, subtraction, multiplication

4. (5 points) The time taken for a text or email message to travel from Atlanta to Seattle is best expressed in units of

(a) minutes to hours

(b) Gigabits/second

(c) Megabytes

(d) milliseconds to seconds

(e) picoseconds to nanoseconds

5. (5 points) What is printed by the Python code?

x = 5

y = x + 3

x = x - 1

z = 10

x = x + z

print(x, y,z)

6. (5 points) What is printed by the Python code?

print(14//4, 14%4, 14.0/4)

7. (5 points) What is printed by the Python code?

n = 3 #1

for x in range(3): #2

n = n + x #3

print(n) #4

8. (5 points) What is printed by the Python code?

def f1():

print('Hi')

def f2():

print('Lo')

f2()

f1()

f1()

9. (10 points) Suppose you are given a list of words, *wordList*. Write Python code that will write one line for each

word, repeating that word twice. For example if *wordList* is ['Jose', 'Sue', 'Ivan'], then your code would print the output shown below, but of course, the code should work for any *wordLiist* irrespective of number of elements

Jose Jose

Sue Sue

Ivan Ivan

10. (10 points)

1. Write a Python class Employee with attributes emp\_name, emp\_id, emp\_salary, and emp\_department and methods calculate\_emp\_salary, emp\_assign\_department, and print\_employee\_details. Below are some sample Employee Data:  
   "ADAMS", "E7876", 50000, "ACCOUNTING"  
   "JONES", "E7499", 45000, "RESEARCH"  
   "MARTIN", "E7900", 50000, "SALES"  
   "SMITH", "E7698", 55000, "OPERATIONS"

* Use 'assign\_department' method to change the department of an employee.
* Use 'print\_employee\_details' method to print the details of an employee.
* Use 'calculate\_emp\_salary' method takes two arguments: salary and hours\_worked, which is the number of hours worked by the employee. If the number of hours worked is more than 50, the method computes overtime and adds it to the salary. Overtime is calculated as following formula:

overtime = hours\_worked – 50  
overtime amount = (overtime \* (salary / 50))

1. Show the output when the following program runs

employee1 = Employee("ADAMS", "E7876", 50000, "ACCOUNTING")

employee2 = Employee("JONES", "E7499", 45000, "RESEARCH")

employee3 = Employee("MARTIN", "E7900", 50000, "SALES")

employee4 = Employee("SMITH", "E7698", 55000, "OPERATIONS")

print("Original Employee Details:")

employee1.print\_employee\_details()

employee2.print\_employee\_details()

employee3.print\_employee\_details()

employee4.print\_employee\_details()

# Change the departments of employee1 and employee4

employee1.assign\_department("OPERATIONS")

employee4.assign\_department("SALES")

# Now calculate the overtime of the employees who are eligible:

employee2.calculate\_salary(45000, 52)

employee4.calculate\_salary(45000, 60)

print("Updated Employee Details:")

employee1.print\_employee\_details()

employee2.print\_employee\_details()

employee3.print\_employee\_details()

employee4.print\_employee\_details()

11. (5 points) Draw an approximate diagram of what the following program might display

from graphics import \*

def main():

win = GraphWin("My Exam", 100, 100)

c = Circle(Point(50,50), 10)

p1 = Point(25,25)

p2 = Point(75,75)

l = Line(p1,p2)

c.draw(win)

l.draw(win)

r = Rectangle(p1,p2)

r.draw(win)

win.getMouse() # pause for click in window

win.close()

main()

12. (5 points) Write a function near\_or\_far() that accepts two ints a and b as arguments. Return True

if a and b are "close" (differing from each other by at most 1), or "far", differing from each other by 2 or more.

13. (10 points) Write a function substring\_common() that accepts two strings as arguments. Return the

number of the positions where they contain the same initial substring. For example, "computer"

and "company" should return 4 since the first 4 letters are the same, “hello” and “hi” should return 1 since only the first letter is the same, and “spring” and “winter” should return 0 since the two strings have no common beginnings.

14. (10 points) Write a function *pick* that takes three string parameters named word1, word2, and word3.This function returns a string that is randomly one of three strings: (i) a new word that is word1 followed by word2; or (ii) a new word that is word1 followed by word3; or (iii) a new word that is word2 followed by word3.

For example pick("car", "boat", "train") could return `carboat' or `cartrain' or ‘boattrain’ and as another example, pick('blue','red','green') could return `redgreen' or `bluered' or ‘bluegreen’

Complete the function pick below.

import random

def pick(word1, word2, word3):

15. (10 points) Write the function named bigEntry that has one parameter, a dictionary named dictnums, where each key is a string mapped to an integer. This function should return the key whose value is the largest. For example, assume the dictionary d is the following: dictnums = {"B": 6, "J": 9, "R":3, "Y":8} the call bigEntry(dictnums) should return “J”, i.e. the key whose corresponding value is the largest in the dictionary.