Answer Key:

FINAL EXAM, VERSION 1 CSci 127: Introduction to Computer Science Hunter College, City University of New York

Spring 2025

- 1. (a) What will the following Python code print: mon_s = "January-February-March-April-May-June" months = mon_s.split('-') print(len(months), "months") print("Last month is", months[-1]) short = [mo[:1] for mo in months] mess = short[-1]print("Short is:", short) firsts = {} for s in short: if s in firsts: firsts[s] = firsts[s]+1 else: firsts[s] = 1print("Months with A:", firsts['A']) print("Months with J:", firsts['J']) Answer Key: 6 months Last month is June Short is: ['J', 'F', 'M', 'A', 'M', 'J'] Months with A: 1 Months with J: 2
 - (b) Consider the following shell commands:

```
$ ls
hello.cpp p1_hello.py p2_triangle.py
$ pwd
/tmp/final/v1
```

Assuming the commands below are run sequentially, what is the output after each has run:

i. \$ mv hello.cpp p1.cpp \$ ls Answer Key:

p2_triangle.py

```
p1_hello.py
                      p1.cpp
   $ mkdir pyprogs
ii. $ mv *.py pyprogs
   $ ls
   Answer Key:
   p1.cpp
                 pyprogs
   $ cd pyprogs
iii. $ echo "Current directory:"
   $ pwd
   Answer Key:
   Current directory:
   /tmp/final/v1/cprogs
   $ mkdir old_files
   $ cp p1.cpp old_files
iv. $ cp program is:"
   $ ls | wc -l
   Answer Key:
   Count is:
   3
```

2. (a) For each question, check all that apply:

Answer Key:

i.	What color is tom after this command? tom.color("#AA0000")?					
	\Box white	\Box green	\Box gray	\checkmark red	\Box blue	
ii.	What is the binar	y number equivale	ent to the decimal	number 17?		
	□ 00111	□ 01001	\Box 10010	✓ 10111	□ 11110	
iii.	Which of the binary numbers below are smaller than the decimal number 9?					
	√ 10	√ 101	□ 1010	□ 1111	\Box none	
iv.	Select the smallest hexadecimal number:					
	\Box AA	□ 31	\square 2C	✓ 1F	\Box FF	
v.	Which of the hexadecimal numbers below are larger than the decimal number 20 ⁴					
	✓ A	✓ F	□ 19	□ 5A	\Box none	

(b) After executing the Python code, write the name of the turtle:

i. which is red:

```
import turtle
ellie = turtle.Turtle()
turtle.colormode(1.0)
ellie.color(0.0, 0.0, 1.0)
fatima = turtle.Turtle()
turtle.colormode(255)
fatima.color(255, 0, 0)
guo = turtle.Turtle()
guo.color("#EFEFEF")
hector = turtle.Turtle()
hector.color("#009999")
```

Answer Key: fatima

ii. which is blue-green:

Answer Key: hector

iii. which is blue:

```
Answer Key: ellie
```

iv. which is gray:

Answer Key: guo

(c) Consider the code:

Answer Key: (i) 1 mess == "" (ii) 2 while mess == "" 3 mess = input('Enter non-empty string: ') 4 print(mess)

The answer should include:

- Mark line 1 with a "(i)".
- In line 1, circle the == (should be =).
- Mark line 2 with a "(ii)".
- At the end of line 5, box the space/parenthesis at the end of the line (where the missing colon should be).

i. Circle the code above and mark line with (i) that caused this error: line 1: mess == ""

```
NameError: name 'mess' is not defined
Write the code that would fix the error:
```

Answer Key:

mess = ""

ii. Box the code above and mark line with (ii) that caused this error: line 2: while mess == ""

SyntaxError: expected ':' Write the code that would fix the error:

while mess == "": 3. (a) What is the value (True/False) of out: in1 = False i. in2 = True out = in1 and in2Answer Key: out = False in1 = False ii. in2 = False out = not in2 or (in2 and not in1) Answer Key: out = True n1 out in2 iii. in1 = False in2 = True Answer Key: out = True

- (b) Fill in the values to yield the output:
 - in1 = i. in2 =

out = in1 and (not in1 or in2)

(c) Design a circuit that implements the logical expression:

(in1 and in2) or not ((in1 and in3) or (in2 and not in3))



4. (a) Draw the output for the function calls:

i. ramble(tim,0)

Answer Key:

import turtle
tim = turtle.Turtle()
tim.shape("turtle")

def ramble(t,side):
 if side < 3:
 t.stamp()
 else:
 for i in range(side):
 t.forward(50)
 t.left(360/side)
 ramble(t,side-1)</pre>



ii. ramble(tim,5)



```
(b) For the following code:
    def v4(antonio, lola):
        if antonio + lola < 10:
            return antonio
        else:
            return -1
```

```
def start():
    jack = 5
    dandan = 20
    kate = v4(jack,dandan)
    return kate
```

i. What are the formal parameters for v4():

Answer Key: antonio, lola

ii. What are the formal parameters for **start()**:

Answer Key: None

iii. What value does start() return:

Answer Key: -1

5. Write a function unique_visitors() that takes a list of 8-digit strings and returns the number of unique strings that occur. For example:

```
ids = ['12345678','11223344','12312323','12345678']
unique_visitors(ids)
```

would return 3 since there are 4 entries but the first and fourth entries are duplicates of each other.

	Libraries:	No additional– just core Python	
Answer Key:	Input:	list of 8-digit strings	
	Output:	number of unique IDs	

Design Pattern:

Answer Key:

 \Box Accumulator \Box Max/Min \checkmark Finding Duplicates \Box Searching **Principal Mechanisms** (select all that apply):

Answer Key:

✓ Single Loop □ Nested Loop ✓ Conditional (if/else) □ Recursion □ Indexing/slicing ✓ Dictionary □ List Comprehension □ Regular Expressions **Process** (as a concise and precise LIST OF STEPS / pseudocode): (Assume libraries have already been imported.)

Answer Key:

(a) Set up an empty dictionary, new_dict.

- (b) For each ID in the ID list:
- (c) Check if the ID is in the dictionary.
- (d) If it is, increment the count
- (e) If it isn't, add ID with value 1 to the dictionary.
- (f) Return the length of the dictionary, new_dict.

- 6. Fill in the Python program that will:
 - prompt the user for the name of a CSV file,
 - prompt the user for the name of a column in that CSV file,
 - print out the maximum value of the column,
 - print out the average value of the column, and
 - displays a plot of the column entered (with "Year" as the x-axis).

Answer Key:

```
#Import the libraries for data frames and displaying images:
import pandas as pd
import matplotlib.pyplot as plt
#Prompt user for file name:
fileName = input('Enter file name: ')
#Prompt user for column name:
col = input('Enter column name: ')
df = pd.read_csv(fileName)
#Compute maximum value of the column:
M = df[col].max()
print("Maximum of column", col, "is", M)
#Compute average value of the column:
ave = df[col].mean()
print("Average of column", col, "is", ave)
#Display a plot of "Year" vs. column entered by user:
df.plot(x = "Year", y = col)
plt.show()
```

- 7. Write a complete Python program that
 - asks the user for the name of a .png (image) file and
 - prints the number of pixels that are very purple (the fraction of red and the fraction of blue are both above 0.75 and the fraction of green is below 0.25).

Answer Key:

```
#Count number of purple pixels in an image
```

#Import the packages for images and arrays: import matplotlib.pyplot as plt

```
import numpy as np
fileName = input('Enter file name: ')
img = plt.imread(fileName) #Read in image
count_purple = 0 #Number of pixels that are purple
#For every pixel:
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        if (img[i,j,0] > 0.75) and (img[i,j,1] < 0.25) and (img[i,j,2] > 0.75):
            count_purple = count_purple + 1
```

```
print("Purple count is", count_purple)
```

8. (a) Consider the following MIPS program:

```
ADDI $s0, $zero, 1
ADD $s1, $s0, $s0
ADD $s2, $s1, $s0
SUB $s3, $s1, $s2
```

After the program runs, what is the value stored in:

\$s1 register	\$s2 register	\$s3 register	
Answer Key: 2	Answer Key: 3	Answer Key: -1	

(b) Consider the MIPS code:

```
1 ADDI $sp, $sp, -6
2 ADDI $t0, $zero, 65
3 ADDI $s2, $zero, 75
4 SETUP: SB $t0, 0($sp)
5 ADDI $sp, $sp, 1
  ADDI $t0, $t0, 2
6
  BEQ $t0, $s2, DONE
7
8 J SETUP
9 DONE: ADDI $t0, $zero, 0
<sup>10</sup> SB $t0, 0($sp)
11 ADDI $sp, $sp, -5
12 ADDI $v0, $zero, 4
  ADDI $a0, $sp, 0
13
   syscall
14
```

i) How many characters are printed?	5
ii) What is the first character printed?	Α
iii) What is the whole message printed?	ACEGI
iv) Detail the changes needed to the code	Line 2: Start t0 at 73.
to print the message in reverse:	Line 3: Start s2 at 63.
	Line 6: Subtract 2 from t0.

9. (a) What is the output

```
//Neil deGrasse Tyson
#include <iostream>
using namespace std;
int main()
{
    cout << "There is no "
        << "greater educ";
    cout << "ation\nthan one ";
    cout << "that is self-driven."
        << endl;
}</pre>
```

Answer Key:

There is no greater education than one that is self-driven.

(b) What is the output:

```
#include <iostream>
using namespace std;
int main()
ſ
    int year=1, bal=1000, expenses=200;
   while( bal > 0 ) {
        cout << "Year " << year
            << ": Balance: $"
            << bal << endl;
       bal = bal - expenses;
       year++;
   }
   return 0;
}
Answer Key:
Year 1: Balance: $1000
Year 2: Balance: $800
Year 3: Balance: $600
Year 4: Balance: $400
```

Year 5: Balance: \$200

(c) What is the output:

```
#include <iostream>
using namespace std;
int main(){
    for (int i=0; i<5; i++){</pre>
         for(int j=0; j<5; j++){</pre>
             if ((i+j) % 2 == 0)
                  cout<<"+";
             else
                   cout<<"-";
          }
          cout << endl;</pre>
     }
      return 0;
}
Answer Key:
+_+_+
_+_+_
+_+_+
_+_+_
+_+_+
```

10. (a) Translate the C++ program into a **complete** Python program:

```
C++ program:
```

```
#include <iostream>
using namespace std;
                                               Python program:
int main()
{
  int num = 1;
                                               Answer Key:
  while ((num < 0) || (num \%2 == 1))
                                               num = 1
  {
                                               while (num > 100) or (num \% 2 == 1):
    cout << "Enter small even #:";</pre>
                                                   num = int(input("Enter small even #: "))
    cin >> num;
                                               print("Your number:", num)
  }
  cout << "Your number: " << num;</pre>
  return 0;
}
```

(b) Write a C++ program that will ask for the time in 24 hour format (e.g. 2034 is 8:34pm) and, prints out "Morning Twilight" if the time is between 5am (e.g. 500) and 5:45am (e.g. 545), "Daylight" if the time is between 5:45am (e.g. 545) and 8pm (e.g. 2000) "Evening Twilight" if the time is between 8pm (e.g. 2000) and 8:30pm (e.g. 2030), and otherwise print "Night" A sample run:

Enter time: 1750 Evening Twilight

Answer Key: #include <iostream> using namespace std; int main() { int time; cout << "Enter time: ";</pre> cin >> time; if ((500 < time) && (time < 545)) { cout << "Morning Twilight \n";</pre> } else if ((545 < time) && (time < 2000)){ cout << "Daylight \n";</pre> } else if ((2000 < time) && (time < 2030)){ cout << "Evening Twilight \n";</pre> } else { cout << "Night \n";</pre> } return 0; }