**CFRS 772: Forensic Artifact Extraction**

**Homework Project 1**

1. **Confirm Python on the lab system**

- Start:Search:python

or

- Install Python 3.x if needed (e.g., personal computer)

1. **Open terminal**

- run "IDLE (Python 3.x GUI)" or similar

- you do not need to run this as Administrator

1. **Check python version installed:**

- IDLE window (before the Python command prompt) should start with "Python 3.x.x"

or

>>> import sys

>>> print(sys.version)

1. **Set command history in IDLE:**

- Options:ConfigureIDLE:Keys

- Scroll down and select "history-next"

- Click on "Get new keys"

- Scroll down to "Down Arrow" and click OK

- Name the Custom Key Set "CFRS772"

- Repeat for "history-previous" with "Up Arrow"

- Apply, OK

1. **Basic interaction:**

- Which of the below work? (put y/n in the blanks)

>>> print("hello") \_\_\_\_

>>> print('hello') \_\_\_\_

>>> print("hello') \_\_\_\_

>>> print(`hello`) \_\_\_\_

>>> print "hello" \_\_\_\_

>>> print 'hello' \_\_\_\_

- Try the following commands: (put result in the blanks)

>>> print(x) \_\_\_\_

>>> x=10

>>> print(x) \_\_\_\_

>>> print(X) \_\_\_\_

>>> y=3

>>> print(y) \_\_\_\_

>>> print(x+y) \_\_\_\_

>>> print(x\*y) \_\_\_\_

>>> print(x/y) \_\_\_\_

>>> print(x-y) \_\_\_\_

>>> print(x//y) \_\_\_\_

>>> print(x%y) \_\_\_\_

>>> z=x+y

>>> print(z) \_\_\_\_

1. **Basic script creation:**

- In IDLE, go to File:New File

- Enter the following in the Editor Window:

x=10

y=3

print(x+y)

- File:SaveAs

test1.py (note the location)

- Run:RunModule or F5

check IDLE console for output

- In console

Try:

>>> test1.py

Did this work?

Try:

>>> exec(open("test1.py").read())

**TAKE SCREENSHOT (alt-PrintScreen, Snipping Tool, etc. and paste into document)**

- At Windows command line

Start:Search:cmd

Run "cmd.exe" (not as Administrator)

Change to directory c:\python3x

C:\Python3x>python <path>/test1.py

**TAKE SCREENSHOT**

1. **Comments and conditionals:**

- In IDLE, File:Open

test1.py

- Add comments:

# This is a one-line comment

''' This is a

multiline comment'''

- Add an IF statement to the end of test1.py:

// The if statement contents, i.e., the stuff after the

// "if" statement, \*must\* be indented (use tab key),

// like this:

if x==10:

print ('x=10')

- Run the script

**TAKE SCREENSHOT**

- Edit the script to have only this content:

x=10

if x==10:

print ('x=10')

elif x==9:

print ('x=9')

else:

print ('x is not 9')

print ('x is not 10')

print ('This line should always print')

- Run the script

**TAKE SCREENSHOT**

- Change x=8 and run the script

**TAKE SCREENSHOT**

- Change x=9 and run the script

**TAKE SCREENSHOT**

1. **Explore context:**

- Start a new IDLE session

>>> print(x)

>>> x=7

>>> print(x)

Run the test1.py script (it should have "x=9" in it)

>>> print(x)

- What does this result tell you? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> x=6

>>> print(x)

Remove the "x=..." line from your test1.py script

Run the test1.py script

Why didn't that work? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>> x=10

>>> print(x)

Run the test1.py script

- Why didn't that work? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Create a Python script**

Create a new file (script) in IDLE; name the file hw1\_yourname.py

Create a simple program with a main routine that calls two functions, add() and subtract(); for each function, include comments about its purpose but just print a short message and return

**TAKE SCREENSHOT**

Prompt for user input in the main routine; create three prompts: number 1, number 2, and operation

**TAKE SCREENSHOT**

Add code (IF block) to the main routine to check the value of the 3rd user argument and call the appropriate function with the first two user arguments

**TAKE SCREENSHOT**

Modify the two functions so they take two arguments (the two numbers entered by the user) and compute and return the sum or difference; add try-except structures to catch errors

**TAKE SCREENSHOT**

Modify the main routine to print out the user input and result

**TAKE SCREENSHOT**

Run a few different examples, including ones that generate errors

**TAKE SCREENSHOT**

Add code to use the debugger; run your script in debugging mode, step through your program, and check values at various points

**TAKE SCREENSHOT**

**On BlackBoard, submit (1) a single PDF document with all screenshots (numbered) and answers to questions (all the blanks, also numbered; you can insert answers into this document and save it as PDF for submission), and (2) your final code for part 9 with debugging disabled.**