CPSC 231: Extra final exam review questions

Extra-extra review questions: They are meant to provide you with some extra practice so you need to actually try them on your own to get anything out of it. For that reason, solutions won't be posted and I won't just "email you the solutions". But it really shouldn't be a big loss because I have taught you how you can find out the answers for yourself:

- For the 'tracing' questions where you have to determine the output of a program then you can always type in the program or expression and run it yourself. (Copy and paste works well out of PDF files).
- The 'writing' questions (e.g. write a query or write a program) are a little trickier but you can still check your answers by testing the results e.g. does your program produce the results specified in the question.
- If you are still unsure of things after you have attempted a question (e.g. you can't figure out why you got a given result after typing it in and running the program), then you can ask (just be sure to show me the work that you have done so I can see how far that you have gotten and perhaps where you are having problems). Again, don't just coming in cold without making an attempt.

Short Answer #1

What is the output of the following program?

```
a = 1
b = 2

def fun ():
  a = 10
  b = 20

def fun1 (a,b,c,d,e):
  a = a + a
  b = b + b
  c = c + c
  d = ['x','y','z']
  e = "bar"

def fun2 (d,e):
  d[0] = "X"
  d[1] = "Y"
  d[2] = "Z"
  e = e + "c"
  return(e)

def start ():
  b = 3
  c = 4
  d = [10,'a',"foo"]
  e = "cps"
  print(a, b, c, d, e)
  fun ()
  print(a, b)
  fun1(a,b,c,d,e)
  print(a, b, c, d, e)
  fun2(d,e)
  print(d, e)

start()
# write your answer here
```

Short Answer #2

Write a function called ‘`findHighest()`’ that will find and return the highest grade in the list called ‘`grades`’ in the following program. You can assume that the list has already been properly initialized. The number of elements in the list is determined by the constant called ‘`SIZE`’.

```
import random

SIZE = 10

def display(grades):
    for i in range (0, SIZE, 1):
        print(grades[i])

def initialize(grades):
    for i in range (0, SIZE, 1):
        grades.append(random.randrange(1,101))


def findHighest (grades):
        # Write your answer here









        # End answer space
    return(highest)

def start():
    grades = []
    initialize(grades)
    display(grades)
    highest = findHighest(grades)
    print('highest', highest)

start()
```

Short Answer #3

For the following program write a function called '`fileWrite()`' that will take the list '`grades`' as a parameter. It will write the grades to an output file called '`data.txt`'. Each grade will reside on its own line. You can assume that the constant `SIZE` will be used to determine the number of elements in the list.

```
SIZE = 10

def initialize():
    # Assume it creates the list properly and returns a
    # reference back to the caller properly.
    return(grades)

# Begin function definition
```

```
# End of answer space

def start():
    grades = initialize()
    fileWrite(grades)

start()
```

(Rough work space)

Short Answer #4

Write a function ‘swap()’ that takes two numbers as parameters. This function will swap what’s currently stored in the numbers and return the two numbers back to its caller. You can add new statements to the swap function but you can’t delete or rewrite the existing statements below.

```
def swap (num1, num2):
    # Insert answer here







    return(num1,num2)

def start():
    num1 = int(input("First number: "))
    num2 = int(input("second number: "))
    num1, num2 = swap (num1, num2)
    # Numbers are swapped now

start()
```

Short Answer #5

Write two functions that perform file output.

`File1(num)` takes an integer as a parameter. It will write to a file the values 1 to that integer (inclusive) to a file each number on a separate line.

```
     # Insert answer here
```

`File2(num)` takes an integer as a parameter. It will write to a file the values 0 to that integer (exclusive) to a file with all numbers on the same line and numbers will be separated by a comma.

```
        # Insert answer here
```

Short Answer #6

What is the output of the following program:

```
# Program: file_trace.py
f = open("data.txt","r")
i = 1
for l in f:
    if ((i % 2) == 0):
        a = l.split(',')
    else:
        a = l.split()
    print(a)
    i = i + 1
f.close()
```

# Input file: input.txt

```
Blue sky to forever
The green grass blows in the wind, dancing
It would be a much better sight with you, with me
If you hadn't met me, I'd be fine on my own, baby
Never felt so lonely, then you came along
```

- **Akira Yamaoka (Silent Hill 3 lyrics)**

# **Output**

Short Answer #7

What is the output of the following program (assume that there is only one file in the same directory as the program which is called “`input.txt`” and the user doesn’t enter a path) if the user enters: (a) `input.txt` (b) `dat.txt`

```
# Program
ok = False
while (ok == False):
    try:
        fn = input("Name of input file: ")
        fv = open(fn,"r")
    except IOError:
        print("Can't open ", fn)
    else:
        print("Opening ", fn)


for l in fv:
    print("%s\t" %(l))

fv.close()
```

```
# Input.txt
Nous sommes des dégourdis,
Nous sommes des lascars
Des types pas ordinaires.
Nous avons souvent notre cafard,
Nous sommes des légionnaires.
Au Tonkin, la Légion immortelle
À Tuyen-Quang illustra notre drapeau,
Héros de Camerone et frères modèles
```

- **Common marching cadence (French foreign legion)**

```
# Program output
```

Short Answer #8

For this question refer to the definition of class "`Ninja`" and the "`start()`" function.

```
class Ninja:
    lives = -1
    clan = ""
    specialPower = ""

def start():
    sho = Ninja(9,"Kuji-kuri","Invisibility")

    # Make a copy
    cho = sho.clone()

    # This only changes the attributes of the copy, the
    # original is unchanged. (Hint: it's not a ninja-magic
    # special ability; just take advantage of the fact that
    # the clone() method will return a new object. The attributes
    # will be an exact copy of the original but there are 2
    # separate ninja objects with their own attributes. Changing
    # one will not change the other. Also keep in mind that when
    # ever the constructor is called a new object will be returned
    # with the attributes set to the values passed into it.)
    cho.clan = "Kanakuna"
    cho.clan = "Kanakuna"
    cho.power = "Yogen"

start()
```

Part (a):
Write an '`init()`' constructor that will take 3 parameters when called. The method will set the 'lives', 'clan' and 'power' attributes to the values of these three parameters.

Part (b):
Write a '`clone()`' method. This method will create a copy of the ninja whose clone method has been called (same values for all attributes but separate object is created and returned by this method). In the '`start()`' function 'sho' and 'cho' refer to two separate Ninja objects. When the cho object is modified, the attributes of sho object retain their original values.

Short Answer #9

There is an error in the logic of the '`display()`' function. This function is to display a bounding line above, below, left and right of each list element but doesn't. Identify and fix the logic error or errors.

```
SIZE = 8

def createListFileRead():
    r = -1
    c = -1
    world = []
    try:
        inputFile = open("input.txt","r")
        r = 0
        for line in inputFile:
            world.append([])
            c = 0
            for ch in line:
                if (c < SIZE):
                    world[r].append(ch)
                c = c + 1
            r = r + 1
        inputFile.close()
    except IOError:
        print("Error reading from input.txt")
    return(world)

def display(world):
    for r in range (0, SIZE, 1):
        for i in range (0, SIZE, 1):
            print(" -", end="")
        print()
        for c in range (0, SIZE, 1):
            # Vertical bar before displaying each element
            print("|" + world[r][c], end="")
        print()

    print()

def start():
    world = createListFileRead()
    display(world)

start()
```

Multiple choice

1. The “Agile” development technique can be best classified under which area of computer science?
   a. Human-Computer interaction
   b. Artificial Intelligence
   c. Computer Vision
   d. Software Engineering
   e. Computer security

2. When was the first microprocessor produced?
   a. 1950’s (1950- 1959)
   b. 1960’s (1960 – 1969)
   c. 1970’s (1970 – 1979)
   d. 1980’s (1980 – 1989)
   e. The microprocessor is something that could theoretically exist but its existence depends upon a tremendous break-through in the area of Quantum computing.