**Justin Davis**

**3/9/2020**

**CS 161**

**CS 161 Python Final Study Exercises**

1. What does the following code output?

a\_list = ['10', '20', '30', 'Western Oregon University', 'Go Wolves'] print(a\_list[-2:] + [2019, 2020])

[‘Western Oregon University’, ‘Go Wolves’, 2019, 2020]

1. Given a statement, determine if it’s valid or invalid.

|  |  |
| --- | --- |
| **Statement** | **Valid or Invalid** |
| num\_legs = 4 \* num\_cats | Valid if both are defined |
| 2 \* y = x | Invalid |
| egg\_count = dozen \* 12 | Valid |
| phone\_cost – discount = price | Invalid |

1. With the logic block shown below, what will be the output when rating is assigned with the value 7?   
     
   If rating < 5  
        Put "Poor" to output  
   Else If rating < 6  
        Put "Below Average" to output  
   Else If rating < 7  
        Put " Average " to output  
   Else If rating < 8  
        Put "Above Average" to output  
   Else If rating <= 10  
        Put "Excellent" to output  
   Else  
        Put "Invalid rating" to output

**Above Average**

1. In the following code:
   1. The variable ***cost*** is the function’s \_\_\_**parameter**\_\_\_\_\_
   2. The variable ***c*** is the function call’s \_\_\_**argument**\_\_\_\_\_

def print\_price(cost):  
 price = cost \* 12  
    print(price)

c= input('Enter the cost per item: ')

print\_price(c)

1. What is the value of test\_val after the following code is executed?

a = 10  
test\_val = 5  
if a \* 2 == test\_val:  
    a = a + 3  
else:  
    test\_val = 2 \* a  
test\_val = a + 4

**14**

1. What is the name of the word that is part of the Python language that can’t be used as a variable name?\_\_\_\_\_\_**Keyword**\_\_\_\_\_\_
2. What is the output of the following code?

my\_list = ['banana', 'apple', 'grape', 'fig', 'lemon']  
my\_list.sort()  
print(my\_list.pop(4))

**lemon**

1. Write a statement to calculate the average of a list of integers using the sum() and the len() functions.

average = **sum(int\_list)/len(int\_list)**

1. Given a = 8 and b =24, write an inequality to express the relationship of **a** to **b**. For example: x >= y

**a < b**

1. def cube(x, y, z):  
       print(x \* y \* z)

Write a function call to the cube() function.

**cube(5,3,20)**

1. What does immutable mean?

**The object or variable can’t be changed with out making a new copy with modified values.**

1. What does mutable mean?

**The current object or variable can be changed without copying and destroying the original.**

1. What is the output?

new\_list = [['WOU', 'is',’the’,’best’], 1, 2020,’Wolfie']  
print(new\_list[0])

print(new\_list[0:2])

**[‘WOU’,’is’,’the’,’best’]**

**[[‘WOU’,’is’,’the’,’best’], 1]**

1. What is the name of the data type used for decimal?

**float**

1. A person can get their driver’s license if they are 16 and they pass the written and driving tests.

Write a Boolean expression so a person can get a driver’s license.

**If age >15 and written == ‘pass’ and driving == ‘pass’:**

**license\_allowed = True**

1. What is polymorphism?

**Polymorphism means an object or variable can be different types**

1. What is the final value of x?

scores = { 'A': 90, 'B': 80, 'C': 70, 'D': 60, ‘F’:50 }  
my\_score = 70  
if my\_score not in scores:  
    x = 1  
else:  
    x= 2  
if 'F' in scores:  
    x = x + 5  
else:  
    x = x + 25

**6**

1. What condition should replace XXX to output "Identical" only if the values of two variables are the same?

my\_word = input("Enter a word: ")  
your\_word = input("Enter a word: ")  
if XXX:  
    print("Identical")

**my\_word == your\_word**

1. What is the output of the following code?

a\_list =[1, 2, 3, 4]  
for i in range(len(a\_list)):  
    if a\_list[i] != a\_list[-1]:  
        a\_list[i] \*= 2  
print(a\_list)

**[2,4,6,4]**

1. List 3 reason for writing functions.
   1. **Easier to maintain code**
   2. **Less repeated code**
   3. **Easier to read the code**
2. What operator would be used? 21\_\_\_\_7 = 3.0 (\*,//,/,%,+,etc)

**/**

1. What operator would be used? 27\_\_\_\_7 = 6

**%**

1. Given x = 11, y = 22, and z = 33,

how is the expression evaluated, True or False?

In the choices, items in parentheses are evaluated first.

(x == 11) or (y == 24) and (z == 33)

**True**

1. What is the value of b\_list?

new\_list = [‘I’, ‘like’, ‘computer’, ‘science’]

b\_list = [i + '#' for i in new\_list[0]]

**[‘I#’]**

1. What is the output?

def number\_of \_students (student\_count):  
    if student\_count < 12:  
        print('Too few.')  
     
    elif (student\_count >= 12) and (student\_count <= 25):  
        print('Perfect size.')  
     
    elif (student\_count > 25):  
        print('Too many.')

number\_of \_students (20)  
number\_of \_students (29)

**Perfect size.**

**Too many.**

1. What is the output?

count = 0  
while count <= 5:  
     print('Wolves')  
count = count + 1  
print('final value of count:', count)

**Infinite loop the count variable doesn’t increment in the while**

1. What is the output?

a1 = [4, 2, 6, 3]  
a1.sort()  
a2 = a1.copy()  
a1.append(10)  
a2 = a2[::-1]  
print(a1, a2)

**[2,3,4,6,10],[6,4,3,2]**

1. Write an equivalent expression to the following:

x = x - 3 if num < 10 else x + 7

**if num < 10:**

**x -= 3**

**else:**

**x += 7**

1. What is the output of the following expression?

print(sys.argv)

**Depends on the computer for the actual but it prints the location of the currently executing file such as:** **['C:/Users/Justin/Desktop/Python Files/Template.py']**

1. What is the value of the \_\_name\_\_ built-in variable in a module that is executed as a script by the programmer?

**‘main’**

1. What statement replaces XXX, causing the program to print the word 'Hurray!'?

def hurray():  
    print('Hurray!')  
  
XXX  
another()

hurray()

1. Write an expression to calculate the square root of a2 + b2?

**Math.sqrt(a\*a+b\*b)**

1. What is the ending value of count?

my\_list = [4, -9, 0, -1, 12, 1, 9]  
x = 0  
count = 0  
  
while x < len(my\_list):  
     if my\_list[x] > 0:  
        count = count + 1  
    x = x + 1

**4**

1. What is the length of the dictionary?

grocery\_dict = {'Product':'Apple', 'Brand': {'Type':'Jonagold', 'Code':123}}

**3**

1. What is the output?

my\_dict = {'First':1, 'Second': 2, 'Third':1, ‘Fourth':0}  
a1 = list(my\_dict.keys())[list(my\_dict.values()).index(1)]  
a2 = {i:j for j, i in my\_dict.items()}[0]  
print(a1,a2)

**First Fourth**

1. What initial value of z will cause an infinite loop?

z = int(input())  
while z != 0:  
    z = z + 4  
    print(z)

**Anything above 0 or below that wont increment by 4 to 0**

1. Write a print statement to display: I can’t wait for break!

**print(‘I can\’t wait for break!’)**

1. Write a print statement to display: 'C:\WOU\Omlid\tests.txt'

(without the single quotes)?

**print(‘C:\\WOU\\Omlid\\tests.txt’) The first two double \\ aren’t needed but it works fits so I left it.**

1. What code replaces XXX to make the program output the number 21?

x = 1  
  
def calc(a):  
 return a \* x  
  
def set\_x(a):  
    XXX  
    x = a  
  
num = 7  
set\_x(3)  
print(calc(num))

**global x**

1. Which expression replaces XXX to make the loop ask for scores until 'quit' is entered?

score = input("Enter a score ('q' to quit)? ")  
while XXX:  
    print('The score is', score)  
    score = input("Enter a score ('q' to quit)? ")

**score != ‘q’**

1. How many times will the body of the loop be executed?

number = 6  
guess = 3  
while number != guess:  
    if number > guess:  
        guess = guess + 1  
    else:  
        guess = guess - 1  
print('The number is:', guess)

**3**

1. What is the output?

mydict = {x: x\*x for x in range(0, 4)}  
for item in mydict.values():  
    print(item, end=' ')

**0 1 4 9**

1. If text = 'WOU is a cool school', what is the value of text[4]?

**i**

1. What is the output?

def foo(items):  
    items = []  
    print(items)  
  
items= [8, 3, 19, 21, 88]

foo(items)  
print(items)

**[]**

**[8,3,19,21,88]**

1. What is the output?

dict1 = {}  
dict1['first'] = 7  
dict1['second'] = 9  
dict1['third'] = 3  
dict1['fourth'] = 1  
dict1['fifth'] = 5

mylist = list(dict1.values())  
sorted\_mylist = sorted(mylist)  
for x in sorted\_mylist:  
    print(x, end=' ')

**1 3 5 7 9**

1. What is the ending value of z?

z = 0  
a = 4  
while a > 1:  
    z = z + a  
    a = a – 1

**9**

1. What is the output?

students = {1: {'student': 'Jon', 'major': 'Physics', 'year': '2018'}, 2: {'student': 'Ann', 'major': 'Math', 'year': '2019'}}  
students[3] = {}  
students[3]['student'] = 'Sara'  
students[3]['major'] = 'Computer Science'  
students[3]['year'] = '2020'  
print(students[3])

**{‘student’:’Sara’, ‘major’:’Coputer Science’, ‘year’: ‘2020’}**

1. What is the output?

names = ['Ann', 'Bill', 'Sam']  
ages = [54, 19, 73]  
for index in [2, 0, 1]:  
    print(names[index] + ":" + str(ages[index]))

**Sam:73**

**Ann:54**

**Bill:19**

1. What are the contents of **pets** after the following code is executed?

pets = [‘Tilly’, ‘Rex’, ‘Abby’]  
ages = [2, 5, 3]  
pets = pets + ages

**[‘Tilly’, ‘Rex’, ‘Abby’, 2, 5, 3]**

1. Use the foo() function to output: **WOU is the best**

def foo(\*args):

    new\_string = ''  
    for item in args:  
        new\_string += ' ' + item  
    return new\_string

**print(foo(‘WOU is the best’))**

1. What is the difference between lists and tuples?

**Lists are changeable while tuples are not**

1. Which XXX / ZZZ outputs every id number/name pair in the dictionary, as in: **123: Alice**?

students = {'123' : 'Alice', '234' : 'Bill', '345' : 'Pam', '456' : 'Erik'}

for XXX:  
    print(ZZZ)

**students = {'123':'Alice', '234':'Bill', '345':'Pam', '456':'Erik'}**

**for id in students:**

**print(id + ':' + students[id])**

1. What is the correlation between namespaces and modules? What happens to namespace when a module is imported?
2. What sequence is generated by range(2, 11, 3)?

**Every third item is displayed from index 2 up to 11**

1. Explain the purpose of the module sys.path.

**Sys.path gets the directory of the current file**

1. What values are in **result\_set** after the following code is run?

a\_set = {1, 3, 5, 7, 9, 11}

b\_set = {3, 5, 7}  
result\_set = a\_set.union(b\_set)

**{1,3,5,7,9,11}**

1. Where is the correct location for docstrings in a function?

**Immediately following the function def()**

1. Which range() function call generates every odd number between 41 and 61 (including both 41 and 61)?

**range(41,62,2)**

1. What is the output?

new\_string = 'Time for a break from homework'  
print(new\_string[0:5])

**Time**

1. What is an environment variable?

**A variable that is stored on the computer and can be accessed by any program on that computer**

1. What is the relationship between a key and value? What data structure has keys and values?

**You use the key to get the value. This structure is called a dictionary.**

1. What is the output?

a\_string = 'Oregon'  
an\_index = 0  
while an\_index != len(a\_string)/2:  
    print(a\_string[an\_index:int(len(a\_string)/2)])  
    an\_index += 1

**Ore**

**re**

**e**

1. When is it best to use a while loop?

**When the number of iterations through the loop isn’t defined.**

1. When is it best to use a for loop?

**When you know the exact amount of times it has to run.**

1. Write a statement to import my\_module1.py.

**Import my\_module1**

1. Write whether the data type is mutable or immutable.

|  |  |
| --- | --- |
| **Data Type** | **Mutable/Immutable** |
| int | Immutable |
| list | Mutable |
| float | Immutable |
| dictionary | Mutable |
| set | Mutable |
| tuple | Immutable |
| string | Immutable |

1. How many times will the print statement execute?

for i in range(2, 4):  
    for j in range(7, 17, 3):  
        print('{:d}. {:d}'.format(i, j))

**8**

1. What is the output?

import os   
print(os.\_\_name\_\_)

**os**

1. What will be output for'{x:.4f}'.format(x = 2.437685087)?

**2.4377**

1. Which data type is the correct choice to store the number of goals a player scores for each game in chronological order?

**list**

1. Which data type is the correct choice to store the names of all the students who are Computer Science majors in no specific order?

**set**

1. Give an example of incremental programming.

**Creating the function definitions but making them print a ‘Fix Me’ line to get the main call working correctly.**

1. What is the output?

my\_string = 'WOU 2020'  
index = my\_string.find('',2)  
print(index)

**2**

1. What does the reload() function do?

**It reloads the currently running file to check for changes without shutting down the program.**

1. What is the output?

num = 8;  
while num <= 13:  
    print(num, end=' ')  
    if num == 10:  
        break  
    num += 1

**8 9 10**

1. Fill in the blanks for the print statement to display 'WOU had 3.5k students in 1995'?

**print('{:s} had {:.1f}k students in {:d}'.format(‘WOU', 3.5, 1995))**