1. Construct a class hierarchy for bank accounts.

Accounts(Owner, ID, Account number, number of accounts)

* Savings Account(Money, interest, minimum deposit, withdraw fees) – Checking Account(Money) – Credit Account(Money, minimum payments, interest, overspending fees) – Loan account(Money, payments, time until payments start, minimum payments)

1. Explain the difference between a method and a function

A function is a program that when ran always performs the same. Methods are similar to functions, but they modify objects to change based upon their specifications.

3. Modify the following code snippet to report an error if the user enters a string, using try/except:

num = input()

def factorial(n):

if n == 0:

return 1

else:

Try

return n \* factorial(n-1)

Except for an error of a non-number and rerun the program either asking for a new number of trying to run it as either and Int or a float.

factorial(num)

1. Use Psuedocode to devise an algorithm for playing Sudoku

Scan numbers

Work from (1-9)

Check if an #’s can be placed based off of the rules that

1. You cannot have 2 of the same number in a row
2. You cannot have 2 of the same number in a column
3. You cannot have 2 of the same number in a 3x3 cell
4. At the end of the game you should have 9 of each integer (1-9)

5. Use pseudocode to design a class that represents a book.

Create a book with title, authors name, cover page, publisher, miscellaneous information (i.e. illustrators name, publisher, book series)

Functions:

Read a page- read the contents of a page

Write a page – write and save words to a page

Flip page forward – flip to the next page

Flip page backward – flip to the previous page

Open page – open to a specified page

6. Explain the difference between is and == in python

== is used to determine if things are fully and truly equal. This includes even the smallest details like names and such. It is good for checking that things have been modified after running through a function.

Is is used to check if things are similar. They are the same class or have the same values, so like comparing two different turtles for instance. This is good for making sure objects meet certain parameters before advancing in a program

7. Give the Big-O performance of the following code fragment:

for i in range(n):

k = 2 + 2

for j in range(n):

k = 2 + 2

for k in range(n):

k = 2 + 2

N

8. Give the Big-O performance of the following code fragment:

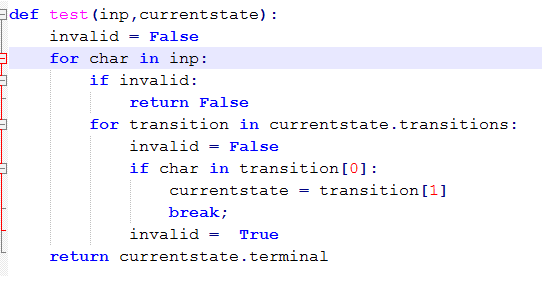
temp\_f = float(input("What is the current temp (F)?"))

print("The temperature in C is " + str((temp\_f - 32 )\* 5/9 ) + "!")

print("The temperature in K is " + str((temp\_f + 459.67 )\* 5/9 )+ "!")

1

9.Give the Big-O runtime of the following code snippet:



1

N

N2

N2

N

N

1

total runtime = 2N2 + 3N + 2

10. Write a python program that uses **recursion** to add up all the digits in a number.

EX: 321 added together would be 6.

def addDigits(num):

   #your solution here

print(addDigits(input("Enter a number: ")))

11.Define the following terms:

a. Big-O notation – The highest possible performance of a program in the respect of runtime performance.

b. Brute force – A program that gets the task done, but not necessarily in a very programming efficient or fast way. These types of programs usually have unnecessary variables and many lines of unnecessary code. Furthermore, these types are programs are usually inefficient and very specific to the task they are programmed to perform.

c. Logarithmic – Breaking down code into smaller pieces that can be more easily computed

d. Complexity (in computer science) – the runtime of programs with respect to efficiency and Big – O notation

LogN – The shortest runtime type possible of any program.

1 – Code that will only run once in a program and isn’t affected by the size of data structures.

N – A loop that will be as long as the specified list is. a.k.a. the program will run once for every item in a list of N.

N2 – Essentially a loop inside a loop. These can rapidly become exponential when cross checking every item in on list with every item in another.

e. Mutability – the ability of a data structure to be changed or modified

f. Superclass – the more widely defined version of a class, used for broad definitions

g. Subclass – the narrower focused class type of a superclass. It is used for more specific things usually.

h. Algorithm – Writing code in plain language, but doing it like you would a program, so broken down line for line of what the program should do.

i. Data structure – an object that stores information

j. Exception – a function that looks for a specific error and is written to perform differently should that error occur

k. Recursion – Using a function that calls upon itself to solve a problem by working its way down to a simple base case for every instance called.