CSE216 – Review Questions

(Note: This is not an exhaustive list of questions. These are given only for reference.)

1. What is the output of the following print() statements?

   print(18 + 3 * 5 ** 3 / 2) _____

   print('Result: ', 56//2 + 100 ** 0.5/4) _____

   print('@*$' * 3 + '%' * 4) _____

   print(eval("51 + (54 * (3 + 2))")) _____

   print(round(10.51)) _____

2. Suppose the following list is defined in an interactive What is the output of each of the following code fragments? There are no syntax errors in the code.

```
>>> a = ["achim", "jeong-o", "jeonyeog", "bam"]
```

   a. `print(len(a))` _____

   b. `print(len(a[1]))` _____

   c. `print(a[2][1:5])` _____

   d. `print(a[3]*3)` _____

   e. `print(a[0] + a[3])` _____

   f. `print(a[1][-3:])` _____

3. Given a variable x that refers to a string as follows, use string slicing to write the following expressions:

```
x = ' Seutobwochie'
```

   a. An expression that is equivalent to the string containing the last 6 characters of x. Your expression should evaluate to `'wochie'`.

      Answer: _____

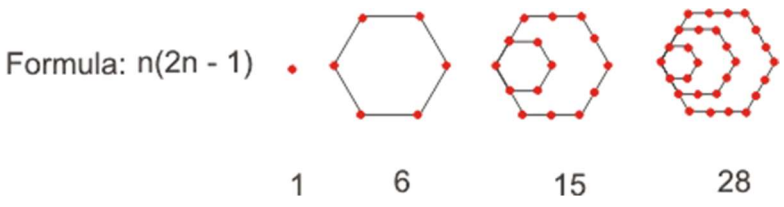
   b. An expression that would extract the word `'Seutob'` from the above string x.

      Answer: _____

   c. An expression that would extract the word `'tobwoc'` from the above string x.

      Answer: _____

4. A **hexagonal numbers series** along with a formula for n^{th} term is given as follows:

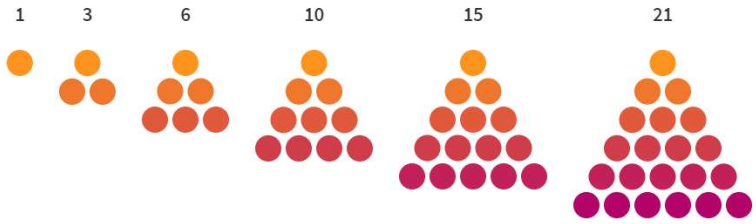


Write a function with the following definition to return a list of n elements in a hexagonal number series where n is the number of terms in a series.

```
def hexagonalSeries(n):
```

```
#testcase
>>> print(hexagonalSeries(6))
[1, 6, 15, 28, 45, 66]
```

5. The **triangle numbers series** is generated by creating triangles of progressively larger size:



Write a function with the following definition to return a list of n elements in a triangle number series where n is the number of terms in a series. E.g. triangleSeries(6) should return a list [1, 3, 6, 10, 15, 21].

def triangleSeries(terms):

6. What is the output of the following?

```
x = ['ab', 'cd']
for i in x:
   x.append(i.upper())
print(x)
```

Answer: _____

7. What is the output of the following?

```
i = 1
while True:
   if i%2 == 0:
   break
   print(i)
   i += 2
```

Answer: _____

8. What is the output when following statement is executed?

```
>>>"abcd"[2:]
```

Answer: _____

9. What is the output when following code is executed?

```
>>> str1 = 'hello'
>>> str2 = ','
>>> str3 = 'world'
>>> str1[-1:]
```

Answer: _____

10. When executed what would the following script print?

```
def func1():
   print(“*”)
   func2()
   print(“@”)
def func2():
   print(“**”)
   func3()
   print(“@@”)
def func3():
   print(“$$”)
def main():
   func2()
   func1()

main()
```

11. Circle appropriate choice. Given a function that does not return any value, What value is thrown by default when executed in shell.

a) int
b) bool
c) void
d) None

12. L = [1, 23, ?, 1]. L is a list. Consider the following items. Which is valid?

a) L = [1, 23, 2, 1]
b) L = [1, 23, ‘a’, 1]
c) L = [1, 23, [1, 23, 1], 1]
d) All of above

13. Which of the following will run without errors (multiple answers possible)?
    a) round(45.8)
    b) round(6352.894,2)
    c) round()
    d) round(7463.123,2,1)

14. Which of the following results in a SyntaxError (Multiple answers possible)?
    a) print('"Once upon a time…", she said.')
    b) print('3”)
    c) print ("'That's okay'")

d) print("He said, "Yes!"")

15. What is the output of print list[2:] if list = ['abcd', 786 , 2.23, 'john', 70.2]?
    a) ['abcd', 786 , 2.23, 'john', 70.2]
    b) abcd
    c) [786, 2.23]
    d) [2.23, 'john', 70.2]

16. Which of the following function convert an integer to a character in python?
    a) set(x)
    b) dict(d)
    c) frozenset(s)
    d) chr(x)

17. What value will be stored in the variable result after the following code has been executed?

```
nums = [6, 5, 4, 8, 10, 12, 5, 7, 0, 9, 11]
result = []
for n in range (int(len(nums) / 2)):
      result.append(nums[n])

      a. [6, 5, 4, 8, 10, 12]
      b. [6, 5, 4, 8, 10]
      c. [6, 5, 4, 8]
      d. None of the above
```

18. What is the output of the following code fragment?

```
name = "Madagascar island"
result = ""
for ch in name:
    if ch.upper() not in 'AS':
          result += ch
print(result)

      a. Madagascar island
      b. Mdgscr islnd
      c. Mdgcr ilnd
      d. Madagacar iland
```

19. What is the output of the following code fragment?

```
d = 10
e = 12
f = 8
if e - f == f - 1:
      print('X')
elif d - f > e:
      print('Y')
else:
      print('Z')

      a. X
      b. Y
      c. Z
      d. None of the above
```

20. Write assignment statements that create dictionaries for the following sets of data:
    • The months of the year, using first three letters of month names as the keys and numbers from 1 to 12 as values.

    • The colors of the rainbow, using the letters in the acronym VIBGYOR as keys and the corresponding colors as values.

```
months = { 'jan': 1, _____
_____ }
colors = _____
_____
```

21. Complete the following function with recursion to convert integer n to a string of binary representation. E.g. dec2bin(15) = 1111

```
def dec2bin(n):
  if n < 0:
    _____
  elif n == 0:
    _____
  else:
    _____
```

22. Write iterative and recursive functions to reverse a list of elements.

23. Write a recursive function to sum elements in a list of numbers.

24. Suppose a variable s has been defined with this assignment statement:
    >>> s = "To be, or not to be, that is the Question:"
    What will Python print for each of the following statements?

    a. >>> print(s) _____
    b. >>> print(len(s)) _____
    c. >>> print(s.split()) _____
    d. >>> import string

       >>> print(s.strip(string.punctuation)) _____

25. Suppose a dictionary object is defined with the following statement:
    >>> d = {'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1}
    What will Python print as the value of the following expressions?

    a. >>> len(d) _____
    b. >>> d['X'] _____

c. >>> d['Z'] _____

d. >>> 'Q' in d _____

e. >>> 5 in d _____

f. >>> list(d.keys()) _____

26. (Classes and OOP)

Create a class called Worker. Worker holds information on a factory worker in a company. The information includes the worker's full name, hourly rate, hours in a standard week and hours in an extended week.
A worker earns their normal hourly rate for the number of hours in a standard week. If they work more hours, for the extra hours, they earn 1.5 times their hourly rate. Finally, if they work beyond the number of hours in the extended week, any hours over that number are paid at 2 times the hourly rate.
The class must have an __init__ method to build the object given the worker's name, hourly rate, standard hours, and extended hours.
You must also write a calculatePay() method that takes the number of hours worked that week and returns the amount of pay in US dollars.

Example: If Joe Cool has a standard work week of 40 hours, an extended week of 50 hours, and wage of 18.50 per hour, his pay for 55 hours would be:

40 * 18.50 + 10 * 18.50 * 1.5 + 5 * 18.50 * 2 = 1202.50

So creating a Worker to compute this would look like:

w = Worker ("Joe Cool", 40, 50, 18.50)
print (w.name + " earned $" + str(w.calculatePay(55)) + " for 55 hours.")

Write the class along with the constructor and the calculatePay() method.

```
class Worker:
```

27. (Code analysis)

What does the following code print:

```
contractions = {"I'm": "I am", "You're": "You are", "He's": "He is",
"She's": "She is"}
sentences = ["I'm finished.", "You're good.", "He's there.", "She's
awesome."]
for sentence in sentences:
    words = sentence.split()
    if words[0] in contractions.keys():
        newsentence = contractions[words[0]]
        for word in words[1:]:
            newsentence = newsentence + " " + word
            print(newsentence)
```

28. Write a password generator in Python where the password is 8 characters long. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and

symbols. The passwords should be random, generating a new password every time the user asks for a new password. Hint: gYou can randomly select an uppercase character using a function random.choice ("ABCDEFGHIJKLMNOPQRSTUVWXYZ")

29. Consider the following class Guitar. Write output of each of the print statements following the class. (15 pts)

```
class Guitar:
    # Construct a guitar object
    def __init__(self, id, numStrings = 12, price = 1000):
        self.id = id
        self.numStrings = 12
        self.price = price

    def getId(self):
        return self.id
    def getNumStrings(self):
        return self.numStrings
    def updatePrice(self, price):
        self.price = price
    def getPrice(self):
       return self.price
    def __str__(self):
        return "Guitar: numStrings = " + str(self.numStrings) + " price = " +
str(self.price) + " id = " + str(self.id)

guitarA = Guitar(123)
print(guitarA)

guitarB = Guitar(234, 8)
print(guitarB)

guitarC = Guitar(345, 8, 1500)
print(guitarC)

guitarD = Guitar(456, price = 2000)
print(guitarD)

guitarA.updatePrice(2500)
print(guitarA)
```

30. Define a Python base class Dog. The Dog class has a constructor method __init__ that accepts two arguments – name and age; and assigns name and age of dog. The Dog class has an instance method speak which takes an argument sound and the dog makes that particular sound. E.g.

```
>>>tommy = Dog("Tommy", 5)

>>>tommy.speak("Bow bow")

Tommy says Bow Bow
```

31. Create two child classes RusselTerrier and Bulldog both of which extend Dog class defined above. Both of these classes implement an instance method run which takes an argument speed and the dog runs with that speed. E.g.

```
>>>jim = Bulldog("Jim", 12)
```

```
>>>jim.run("slowly")

Jim runs slowly
```

32. Given the following code, show the status of subroutine stack for each of the method call, method execution and method finish/return.

```
public static void main(String[]
args) {
 int i = 5;
 int j = 2;
 int k = max(i, j);

 System.out.println("The maximum
is  " + k);
}
```

```
public static int max(int num1, int
num2) {
 int result;
 if (num1 > num2)
   result = num1;
 else
   result = num2;
 return result;
```

More topics:

Polymorphism

Method overloading

Method overriding

Dynamic binding

C Pointers

Pass by value/pass by reference outputs (Java, Python, C)

Java generics (See https://docs.oracle.com/javase/tutorial/java/generics/QandE/generics-answers.html) :

1. Will the following class compile? If not, why?

```
public final class Algorithm {
    public static <T> T max(T x, T y) {
        return x > y ? x : y;
    }
}
```

Answer: No. The greater than (>) operator applies only to primitive numeric types.

2. Write a generic method to exchange the positions of two different elements in an array.

Answer:

```
public final class Algorithm {
    public static <T> void swap(T[] a, int i, int j) {
        T temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }
}
```

3. Write a generic method to find the maximal element in the range [begin, end) of a list.

   Answer:

```
import java.util.*;

public final class Algorithm {
    public static <T extends Object & Comparable<? super T>>
        T max(List<? extends T> list, int begin, int end) {

        T maxElem = list.get(begin);

        for (++begin; begin < end; ++begin)
            if (maxElem.compareTo(list.get(begin)) < 0)
                maxElem = list.get(begin);
        return maxElem;
    }
}
```

4. Will the following class compile? If not, why?

```
public class Singleton<T> {

    public static T getInstance() {
        if (instance == null)
            instance = new Singleton<T>();

        return instance;
    }

    private static T instance = null;
}
```

   Answer: No. You cannot create a static field of the type parameter `T`.

5. Given the following classes:

```
class Shape { /* ... */ }
class Circle extends Shape { /* ... */ }
class Rectangle extends Shape { /* ... */ }

class Node<T> { /* ... */ }
```

   Will the following code compile? If not, why?

```
   Node<Circle> nc = new Node<>();
   Node<Shape>  ns = nc;
```

   Answer: No. Because `Node<Circle>` is not a subtype of `Node<Shape>`.

6. Consider this class:

```
class Node<T> implements Comparable<T> {
    public int compareTo(T obj) { /* ... */ }
    // ...
}
```

   Will the following code compile? If not, why?

   Answer: Yes.

```
Node<String> node = new Node<>();
Comparable<String> comp = node;
```