1. Which of the following statements create a dictionary?  
a) d = {}  
b) d = {“john”:40, “peter”:45}  
c) d = {40:”john”, 45:”peter”}  
d) All of the mentioned  
View Answer

Answer: d  
Explanation: Dictionaries are created by specifying keys and values

2. Read the code shown below carefully and pick out the keys?

1. d = {"john":40, "peter":45}

a) “john”, 40, 45, and “peter”  
b) “john” and “peter”  
c) 40 and 45  
d) d = (40:”john”, 45:”peter”)  
View Answer

Answer: b  
Explanation: Dictionaries appear in the form of keys and values.

3. What will be the output?

1. d = {"john":40, "peter":45}
2. "john" in d

a) True  
b) False  
c) None  
d) Error  
View Answer

Answer: a  
Explanation: In can be used to check if the key is int dictionary.

4. What will be the output?

1. d1 = {"john":40, "peter":45}
2. d2 = {"john":466, "peter":45}
3. d1 == d2

a) True  
b) False  
c) None  
d) Error  
View Answer

Answer: b  
Explanation: If d2 was initialized as d2 = d1 the answer would be true.

5. What will be the output?

1. d1 = {"john":40, "peter":45}
2. d2 = {"john":466, "peter":45}
3. d1 > d2

a) True  
b) False  
c) Error  
d) None  
View Answer

Answer: c  
Explanation: Arithmetic > operator cannot be used with dictionaries.

6. What is the output?

1. d = {"john":40, "peter":45}
2. d["john"]

a) 40  
b) 45  
c) “john”  
d) “peter”  
View Answer

Answer: a  
Explanation: Execute in the shell to verify.

7. Suppose d = {“john”:40, “peter”:45}, to delete the entry for “john” what command do we use  
a) d.delete(“john”:40)  
b) d.delete(“john”)  
c) del d[“john”].  
d) del d(“john”:40)  
View Answer

8. Suppose d = {“john”:40, “peter”:45}. To obtain the number of entries in dictionary which command do we use?  
a) d.size()  
b) len(d)  
c) size(d)  
d) d.len()  
View Answer

Answer: b  
Explanation: Execute in the shell to verify.

9. What will be the output?

1. d = {"john":40, "peter":45}
2. print(list(d.keys()))

a) [“john”, “peter”].  
b) [“john”:40, “peter”:45].  
c) (“john”, “peter”)  
d) (“john”:40, “peter”:45)  
View Answer

Answer: a  
Explanation: The output of the code shown above is a list containing only keys of the dictionary d, in the form of a list.

10. Suppose d = {“john”:40, “peter”:45}, what happens when we try to retrieve a value using the expression d[“susan”]?  
a) Since “susan” is not a value in the set, Python raises a KeyError exception  
b) It is executed fine and no exception is raised, and it returns None  
c) Since “susan” is not a key in the set, Python raises a KeyError exception  
d) Since “susan” is not a key in the set, Python raises a syntax error  
View Answer

Answer: c  
Explanation: Execute in the shell to verify.

1. Which of these about a dictionary is false?  
a) The values of a dictionary can be accessed using keys  
b) The keys of a dictionary can be accessed using values  
c) Dictionaries aren’t ordered  
d) Dictionaries are mutable  
View Answer

Answer: b  
Explanation: The values of a dictionary can be accessed using keys but the keys of a dictionary can’t be accessed using values.

2. Which of the following is not a declaration of the dictionary?  
a) {1: ‘A’, 2: ‘B’}  
b) dict([[1,”A”],[2,”B”]])  
c) {1,”A”,2”B”}  
d) { }  
View Answer

Answer: c  
Explanation: Option c is a set, not a dictionary.

3. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

for i,j in a.items():

print(i,j,end=" ")

a) 1 A 2 B 3 C  
b) 1 2 3  
c) A B C  
d) 1:”A” 2:”B” 3:”C”  
View Answer

Answer: a  
Explanation: In the above code, variables i and j iterate over the keys and values of the dictionary respectively.

4. What is the output of the following piece of code?

a={1:"A",2:"B",3:"C"}

print(a.get(1,4))

a) 1  
b) A  
c) 4  
d) Invalid syntax for get method  
View Answer

Answer: b  
Explanation: The get() method returns the value of the key if the key is present in the dictionary and the default value(second parameter) if the key isn’t present in the dictionary.

5. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

print(a.get(5,4))

a) Error, invalid syntax  
b) A  
c) 5  
d) 4  
View Answer

Answer: d  
Explanation: The get() method returns the default value(second parameter) if the key isn’t present in the dictionary.

6. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

print(a.setdefault(3))

a) {1: ‘A’, 2: ‘B’, 3: ‘C’}  
b) C  
c) {1: 3, 2: 3, 3: 3}  
d) No method called setdefault() exists for dictionary  
View Answer

Answer: b  
Explanation: setdefault() is similar to get() but will set dict[key]=default if key is not already in the dictionary.

7. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

a.setdefault(4,"D")

print(a)

a) {1: ‘A’, 2: ‘B’, 3: ‘C’, 4: ‘D’}.  
b) None.  
c) Error.  
d) [1,3,6,10].  
View Answer

Answer: a  
Explanation: setdefault() will set dict[key]=default if key is not already in the dictionary.

8. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

b={4:"D",5:"E"}

a.update(b)

print(a)

a) {1: ‘A’, 2: ‘B’, 3: ‘C’}  
b) Method update() doesn’t exist for dictionaries  
c) {1: ‘A’, 2: ‘B’, 3: ‘C’, 4: ‘D’, 5: ‘E’}  
d) {4: ‘D’, 5: ‘E’}  
View Answer

Answer: c  
Explanation: update() method adds dictionary b’s key-value pairs to dictionary a. Execute in python shell to verify.

9. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

b=a.copy()

b[2]="D"

print(a)

a) Error, copy() method doesn’t exist for dictionaries  
b) {1: ‘A’, 2: ‘B’, 3: ‘C’}  
c) {1: ‘A’, 2: ‘D’, 3: ‘C’}  
d) “None” is printed  
View Answer

Answer: b  
Explanation: Changes made in the copy of the dictionary isn’t reflected in the original one.

10. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

a.clear()

print(a)

a) None  
b) { None:None, None:None, None:None}  
c) {1:None, 2:None, 3:None}  
d) { }  
View Answer

Answer: d  
Explanation: The clear() method clears all the key-value pairs in the dictionary.

11. Which of the following isn’t true about dictionary keys?  
a) More than one key isn’t allowed  
b) Keys must be immutable  
c) Keys must be integers  
d) When duplicate keys encountered, the last assignment wins  
View Answer

Answer: c  
Explanation: Keys of a dictionary may be any data type that is immutable.

12. What is the output of the following code?

a={1:5,2:3,3:4}

a.pop(3)

print(a)

a) {1: 5}  
b) {1: 5, 2: 3}  
c) Error, syntax error for pop() method  
d) {1: 5, 3: 4}  
View Answer

Answer: b  
Explanation: pop() method removes the key-value pair for the key mentioned in the pop() method.

13. What is the output of the following code?

a={1:5,2:3,3:4}

print(a.pop(4,9))

a) 9  
b) 3  
c) Too many arguments for pop() method  
d) 4  
View Answer

Answer: a  
Explanation: pop() method returns the value when the key is passed as an argument and otherwise returns the default value(second argument) if the key isn’t present in the dictionary.

14. What is the output of the following code?

a={1:"A",2:"B",3:"C"}

for i in a:

print(i,end=" ")

a) 1 2 3  
b) ‘A’ ‘B’ ‘C’  
c) 1 ‘A’ 2 ‘B’ 3 ‘C’  
d) Error, it should be: for i in a.items():  
View Answer

Answer: a  
Explanation: The variable i iterates over the keys of the dictionary and hence the keys are printed.

15. Execute the following in Python shell?

>>> a={1:"A",2:"B",3:"C"}

>>> a.items()

a) Syntax error  
b) dict\_items([(‘A’), (‘B’), (‘C’)])  
c) dict\_items([(1,2,3)])  
d) dict\_items([(1, ‘A’), (2, ‘B’), (3, ‘C’)])  
View Answer

Answer: d  
Explanation: The method items() returns list of tuples with each tuple having a key-value pair.

1. Which of the statements about dictionary values if false?  
a) More than one key can have the same value.  
b) The values of the dictionary can be accessed as dict[key].  
c) Values of a dictionary must be unique.  
d) Values of a dictionary can be a mixture of letters and numbers.  
View Answer

Answer: c  
Explanation: More than one key can have the same value.

2. What is the output of the following snippet of code?

>>> a={1:"A",2:"B",3:"C"}

>>> del a

a) method del doesn’t exist for the dictionary  
b) del deletes the values in the dictionary  
c) del deletes the entire dictionary  
d) del deletes the keys in the dictionary  
View Answer

Answer: c  
Explanation: del deletes the entire dictionary and any further attempt to access it will throw an error.

3. If a is a dictionary with some key-value pairs, what does a.popitem() do?  
a) Removes an arbitrary element  
b) Removes all the key-value pairs  
c) Removes the key-value pair for the key given as an argument  
d) Invalid method for dictionary  
View Answer

Answer: a  
Explanation: The method popitem() removes a random key-value pair.

4. What is the output of the following snippet of code?

total={}

def insert(items):

if items in total:

total[items] += 1

else:

total[items] = 1

insert('Apple')

insert('Ball')

insert('Apple')

print (len(total))

a) 3  
b) 1  
c) 2  
d) 0  
View Answer

Answer: c  
Explanation: The insert() function counts the number of occurrences of the item being inserted into the dictionary. There are only 2 keys present since the key ‘Apple’ is repeated. Thus, the length of the dictionary is 2.

5. What is the output of the following snippet of code?

a = {}

a[1] = 1

a['1'] = 2

a[1]=a[1]+1

count = 0

for i in a:

count += a[i]

print(count)

a) 1  
b) 2  
c) 4  
d) Error, the keys can’t be a mixture of letters and numbers  
View Answer

Answer: c  
Explanation: The above piece of code basically finds the sum of the values of keys.

6. What is the output of the following snippet of code?

numbers = {}

letters = {}

comb = {}

numbers[1] = 56

numbers[3] = 7

letters[4] = 'B'

comb['Numbers'] = numbers

comb['Letters'] = letters

print(comb)

a) Error, dictionary in a dictionary can’t exist  
b) ‘Numbers’: {1: 56, 3: 7}  
c) {‘Numbers’: {1: 56}, ‘Letters’: {4: ‘B’}}  
d) {‘Numbers’: {1: 56, 3: 7}, ‘Letters’: {4: ‘B’}}  
View Answer

Answer: d  
Explanation: Dictionary in a dictionary can exist.

7. What is the output of the following snippet of code?

test = {1:'A', 2:'B', 3:'C'}

test = {}

print(len(test))

a) 0  
b) None  
c) 3  
d) An exception is thrown  
View Answer

Answer: a  
Explanation: In the second line of code, the dictionary becomes an empty dictionary. Thus, length=0.

8. What is the output of the following snippet of code?

test = {1:'A', 2:'B', 3:'C'}

del test[1]

test[1] = 'D'

del test[2]

print(len(test))

a) 0  
b) 2  
c) Error as the key-value pair of 1:’A’ is already deleted  
d) 1  
View Answer

Answer: b  
Explanation: After the key-value pair of 1:’A’ is deleted, the key-value pair of 1:’D’ is added.

9. What is the output of the following snippet of code?

a = {}

a[1] = 1

a['1'] = 2

a[1.0]=4

count = 0

for i in a:

count += a[i]

print(count)

a) An exception is thrown  
b) 3  
c) 6  
d) 2  
View Answer

Answer: c  
Explanation: The value of key 1 is 4 since 1 and 1.0 are the same. Then, the function count() gives the sum of all the values of the keys (2+4).

10. What is the output of the following snippet of code?

a={}

a['a']=1

a['b']=[2,3,4]

print(a)

a) Exception is thrown  
b) {‘b’: [2], ‘a’: 1}  
c) {‘b’: [2], ‘a’: [3]}  
d) {‘b’: [2, 3, 4], ‘a’: 1}  
View Answer

Answer: d  
Explanation: Mutable members can be used as the values of the dictionary but they cannot be used as the keys of the dictionary.

11. What is the output of the following piece of code?

>>>import collections

>>> a=collections.Counter([1,1,2,3,3,4,4,4])

>>> a

a) {1,2,3,4}  
b) Counter({4, 1, 3, 2})  
c) Counter({4: 3, 1: 2, 3: 2, 2: 1})  
d) {4: 3, 1: 2, 3: 2, 2: 1}  
View Answer

Answer: c  
Explanation: The statement a=collections.OrderedDict() generates a dictionary with the number as the key and the count of times the number appears as the value.

12. What is the output of the following piece of code?

>>>import collections

>>> b=collections.Counter([2,2,3,4,4,4])

>>> b.most\_common(1)

a) Counter({4: 3, 2: 2, 3: 1}).  
b) {3:1}.  
c) {4:3}.  
d) [(4, 3)].  
View Answer

Answer: d  
Explanation: The most\_common() method returns the n number key-value pairs where the value is the most recurring.

13. What is the output of the following piece of code?

>>>import collections

>>> b=collections.Counter([2,2,3,4,4,4])

>>> b.most\_common(1)

a) Counter({4: 3, 2: 2, 3: 1}).  
b) {3:1}.  
c) {4:3}.  
d) [(4, 3)].  
View Answer

Answer: d  
Explanation: The most\_common() method returns the n number key-value pairs where the value is the most recurring.

14. What is the output of the following piece of code?

>>> import collections

>>> a=collections.Counter([2,2,3,3,3,4])

>>> b=collections.Counter([2,2,3,4,4])

>>> a|b

a) Counter({3: 3, 2: 2, 4: 2})  
b) Counter({2: 2, 3: 1, 4: 1})  
c) Counter({3: 2})  
d) Counter({4: 1})  
View Answer

Answer: a  
Explanation: a|b returns the pair of keys and the highest recurring value.

15. What is the output of the following piece of code?

>>> import collections

>>> a=collections.Counter([3,3,4,5])

>>> b=collections.Counter([3,4,4,5,5,5])

>>> a&b

a) Counter({3: 12, 4: 1, 5: 1})  
b) Counter({3: 1, 4: 1, 5: 1})  
c) Counter({4: 2})  
d) Counter({5: 1})  
View Answer

Answer: b  
Explanation: a&b returns the pair of keys and the lowest recurring value.

1. The following piece of code is invalid. True or False?

class demo(dict):

def \_\_test\_\_(self,key):

return []

a = demo()

a['test'] = 7

print(a)

a) True  
b) False  
View Answer

Answer: b  
Explanation: The output of the code is: {‘test’:7}.

2. What is the output of the following code?

count={}

count[(1,2,4)] = 5

count[(4,2,1)] = 7

count[(1,2)] = 6

count[(4,2,1)] = 2

tot = 0

for i in count:

tot=tot+count[i]

print(len(count)+tot)

a) 25  
b) 17  
c) 16  
d) Tuples can’t be made keys of a dictionary  
View Answer

Answer: c  
Explanation: Tuples can be made keys of a dictionary. Length of the dictionary is 3 as the value of the key (4,2,1) is modified to 2. The value of the variable tot is 5+6+2=13.

3. What is the output of the following code?

a={}

a[2]=1

a[1]=[2,3,4]

print(a[1][1])

a) [2,3,4].  
b) 3  
c) 2  
d) An exception is thrown  
View Answer

Answer: b  
Explanation: Now, a={1:[2,3,4],2:1} . a[1][1] refers to second element having key 1.

4. What is the output of the following piece of code?

>>> a={'B':5,'A':9,'C':7}

>>> sorted(a)

a) [‘A’,’B’,’C’].  
b) [‘B’,’C’,’A’].  
c) [5,7,9].  
d) [9,5,7].  
View Answer

Answer: a  
Explanation: Return a new sorted list of keys in the dictionary.

5. What is the output of the following snippet of code?

>>> a={i: i\*i for i in range(6)}

>>> a

a) Dictionary comprehension doesn’t exist  
b) {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6:36}  
c) {0: 0, 1: 1, 4: 4, 9: 9, 16: 16, 25: 25}  
d) {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}  
View Answer

Answer: d  
Explanation: Dictionary comprehension is implemented in the above piece of code.

6. What is the output of the following piece of code?

>>> a={}

>>> a.fromkeys([1,2,3],"check")

a) Syntax error  
b) {1:”check”,2:”check”,3:”check”}  
c) “check”  
d) {1:None,2:None,3:None}  
View Answer

Answer: b  
Explanation: The dictionary takes values of keys from the list and initializes it to the default value (value given in the second parameter). Execute in Python shell to verify.

7. What is the output of the following snippet of code?

>>> b={}

>>> all(b)

a) { }  
b) False  
c) True  
d) An exception is thrown  
View Answer

Answer: c  
Explanation: Function all() returns True if all keys of the dictionary are true or if the dictionary is empty.

8. If b is a dictionary, what does any(b) do?  
a) Returns True if any key of the dictionary is true  
b) Returns False if dictionary is empty  
c) Returns True if all keys of the dictionary are true  
d) Method any() doesn’t exist for dictionary  
View Answer

Answer: a  
Explanation: Method any() returns True if any key of the dictionary is true and False if the dictionary is empty.

9. What is the output of the following code?

>>> a={"a":1,"b":2,"c":3}

>>> b=dict(zip(a.values(),a.keys()))

>>> b

a) {‘a’: 1, ‘b’: 2, ‘c’: 3}  
b) An exception is thrown  
c) {‘a’: ‘b’: ‘c’: }  
d) {1: ‘a’, 2: ‘b’, 3: ‘c’}  
View Answer

Answer: d  
Explanation: The above piece of code inverts the key-value pairs in the dictionary.

10. What is the output of the following piece of code when executed in Python shell?

>>> a={i: 'A' + str(i) for i in range(5)}

>>> a

a) An exception is thrown  
b) {0: ‘A0’, 1: ‘A1’, 2: ‘A2’, 3: ‘A3’, 4: ‘A4’}  
c) {0: ‘A’, 1: ‘A’, 2: ‘A’, 3: ‘A’, 4: ‘A’}  
d) {0: ‘0’, 1: ‘1’, 2: ‘2’, 3: ‘3’, 4: ‘4’}  
View Answer

Answer: d  
Explanation: Dictionary comprehension and string concatenation is implemented in the above piece of code.

11. What is the output of the following piece of code when executed in Python shell?

>>> a=dict()

>>> a[1]

a) An exception is thrown since the dictionary is empty  
b) ‘ ‘  
c) 1  
d) 0  
View Answer

Answer: a  
Explanation: The values of a dictionary can be accessed through the keys only if the keys exist in the dictionary.

12. What is the output of the following piece of code when executed in Python shell?

>>> import collections

>>> a=dict()

>>> a=collections.defaultdict(int)

>>> a[1]

a) 1  
b) 0  
c) An exception is thrown  
d) ‘ ‘  
View Answer

Answer: b  
Explanation: The statement a=collections.defaultdict(int) gives the default value of 0  
(since int data type is given within the parenthesis) even if the keys don’t exist in the dictionary.

13. What is the output of the following piece of code when executed in Python shell?

>>> import collections

>>> a=dict()

>>> a=collections.defaultdict(str)

>>> a['A']

a) An exception is thrown since the dictionary is empty  
b) ‘ ‘  
c) ‘A’  
d) 0  
View Answer

Answer: b  
Explanation: The statement a=collections.defaultdict(str) gives the default value of ‘ ‘ even if the keys don’t exist in the dictionary.

14. What is the output of the following piece of code when executed in Python shell?

>>> import collections

>>> b=dict()

>>> b=collections.defaultdict(lambda: 7)

>>> b[4]

a) 4  
b) 0  
c) An exception is thrown  
d) 7  
View Answer

Answer: d  
Explanation: The statement a=collections.defaultdict(lambda: x) gives the default value of x even if the keys don’t exist in the dictionary.

15. What is the output of the following piece of code when executed in Python shell?

>>> import collections

>>> a=collections.OrderedDict((str(x),x) for x in range(3))

>>> a

a) {‘2’:2, ‘0’:0, ‘1’:1}  
b) OrderedDict([(‘0’, 0), (‘1’, 1), (‘2’, 2)])  
c) An exception is thrown  
d) ‘ ‘  
View Answer

Answer: b  
Explanation: The line of code a=collections.OrderedDict() generates a dictionary satisfying the conditions given within the parenthesis and in an ascending order of the keys.

1. Which of the following functions is a built-in function in python?  
a) seed()  
b) sqrt()  
c) factorial()  
d) print()  
View Answer

Answer: d  
Explanation: The function seed is a function which is present in the random module. The functions sqrt and factorial are a part of the math module. The print function is a built-in function which prints a value directly to the system output.

2. What is the output of the expression:

round(4.576)

a) 4.5  
b) 5  
c) 4  
d) 4.6  
View Answer

Answer: b  
Explanation: This is a built-in function which rounds a number to give precision in decimal digits. In the above case, since the number of decimal places has not been specified, the decimal number is rounded off to a whole number. Hence the output will be 5.

3. The function pow(x,y,z) is evaluated as:  
a) (x\*\*y)\*\*z  
b) (x\*\*y) / z  
c) (x\*\*y) % z  
d) (x\*\*y)\*z  
View Answer

Answer: c  
Explanation: The built-in function pow() can accept two or three arguments. When it takes in two arguments, they are evaluated as: x\*\*y. When it takes in three arguments, they are evaluated as: (x\*\*y)%z.

4. What is the output of the function shown below?

all([2,4,0,6])

a) Error  
b) True  
c) False  
c) 0  
View Answer

Answer: c  
Explanation: The function all returns false if any one of the elements of the iterable is zero and true if all the elements of the iterable are non zero. Hence the output of this function will be false.

5. What is the output of the expression?

round(4.5676,2)?

a) 4.5  
b) 4.6  
c) 4.57  
d) 4.56  
View Answer

Answer: c  
Explanation: The function round is used to round off the given decimal number to the specified decimal places. In this case the number should be rounded off to two decimal places. Hence the output will be 4.57.

6. What is the output of the following function?

any([2>8, 4>2, 1>2])

a) Error  
b) True  
c) False  
d) 4>2  
View Answer

Answer: b  
Explanation: The built-in function any() returns true if any or more of the elements of the iterable is true (non zero), If all the elements are zero, it returns false.

7. What is the output of the function shown below?

import math

abs(math.sqrt(25))

a) Error  
b) -5  
c) 5  
d) 5.0  
View Answer

Answer: d  
Explanation: The abs() function prints the absolute value of the argument passed. For example: abs(-5)=5. Hence , in this case we get abs(5.0)=5.0.

8. What are the outcomes of the functions shown below?

sum(2,4,6)

sum([1,2,3])

a) Error, 6  
b) 12, Error  
c) 12, 6  
d) Error, Error  
View Answer

Answer: a  
Explanation: The first function will result in an error because the function sum() is used to find the sum of iterable numbers. Hence the outcomes will be Error and 6 respectively.

9. What is the output of the function:

all(3,0,4.2)

a) True  
b) False  
c) Error  
d) 0  
View Answer

Answer: c  
Explanation: The function all() returns ‘True’ if any one or more of the elements of the iterable are non zero. In the above case, the values are not iterable, hence an error is thrown.

10. What is the output of the functions shown below?

min(max(False,-3,-4), 2,7)

a) 2  
b) False  
c) -3  
d) -4  
View Answer

Answer: b  
Explanation: The function max() is being used to find the maximum value from among -3, -4 and false. Since false amounts to the value zero, hence we are left with min(0, 2, 7) Hence the output is 0 (false).

1. What are the outcomes of the following functions?

chr(‘97’)

chr(97)

a) a  
Error  
b) ‘a’  
a  
c) Error  
a  
d) Error  
Error  
View Answer

Answer: c  
Explanation: The built-in function chr() returns the alphabet corresponding to the value given as an argument. This function accepts only integer type values. In the first function, we have passed a string. Hence the first function throws an error.

2. What is the output of the following function?

complex(1+2j)

a) Error  
b) 1  
c) 2j  
d) 1+2j  
View Answer

Answer: d  
Explanation: The built-in function complex() returns the argument in a complex form. Hence the output of the function shown above will be 1+2j.

3. What is the output of the function complex() ?  
a) 0j  
b) 0+0j  
c) 0  
d) Error  
View Answer

Answer: a  
Explanation: The complex function returns 0j if both of the arguments are omitted, that is, if the function is in the form of complex() or complex(0), then the output will be 0j.

4. The function divmod(a,b), where both ‘a’ and ‘b’ are integers is evaluated as:  
a) (a%b, a//b)  
b) (a//b, a%b)  
c) (a//b, a\*b)  
c) (a/b, a%b)  
View Answer

Answer: b  
Explanation: The function divmod(a,b) is evaluated as a//b, a%b, if both ‘a’ and ‘b’ are integers.

5. What is the output of the functions shown below?

divmod(10.5,5)

divmod(2.4,1.2)

a) (2.00, 0.50)  
(2.00, 0.00)  
b) (2, 0.5)  
(2, 0)  
c) (2.0, 0.5)  
(2.0, 0.0)  
d) (2, 0.5)  
(2)  
View Answer

Answer: c  
Explanation: See python documentation for the function divmod.

6. The function complex(‘2-3j’) is valid but the function complex(‘2 – 3j’) is invalid. State whether this statement is true or false.  
a) True  
b) False  
View Answer

Answer: a  
Explanation: When converting from a string, the string must not contain any blank spaces around the + or – operator. Hence the function complex(‘2 – 3j’) will result in an error.

7. What is the output of the function shown below?

list(enumerate([2, 3]))

a) Error  
b) [(1, 2), (2, 3)]  
c) [(0, 2), (1, 3)]  
d) [(2, 3)]  
View Answer

Answer: c  
Explanation: The built-in function enumerate() accepts an iterable as an argument. The function shown in the above case returns containing pairs of the numbers given, starting from 0. Hence the output will be: [(0, 2), (1,3)].

8. What are the outcomes of the function shown below?

x=3

eval('x^2')

a) Error  
b) 1  
c) 9  
d) 6  
View Answer

Answer: b  
Explanation: The function eval is use to evaluate the expression that it takes as an argument. In the above case, the eval() function is used to perform XOR operation between 3 and 2. Hence the output is 1.

9. What is the output of the functions shown below?

float('1e-003')

float('2e+003')

a) 3.00  
300  
b) 0.001  
2000.0  
c) 0.001  
200  
d) Error  
2003  
View Answer

Answer: b  
Explanation: The output of the first function will be 0.001 and that of the second function will be 2000.0. The first function created a floating point number up to 3 decimal places and the second function adds 3 zeros after the given number.

10. Which of the following functions does not necessarily accept only iterables as arguments?  
a) enumerate()  
b) all()  
c) chr()  
d) max()  
View Answer

Answer: c  
Explanation: The functions enumerate(), all() and max() accept iterables as arguments whereas the function chr() throws an error on receiving an iterable as an argument. Also note that the function chr() accepts only integer values.

1. Which of the following functions accepts only integers as arguments?  
a) ord()  
b) min()  
c) chr()  
d) any()  
View Answer

Answer: c  
Explanation: The function chr() accepts only integers as arguments. The function ord() accepts only strings. The functions min() and max() can accept floating point as well as integer arguments.

2. Suppose there is a list such that: l=[2,3,4].  
If we want to print this list in reverse order, which of the following methods should be used?  
a) reverse(l)  
b) list(reverse[(l)])  
c) reversed(l)  
d) list(reversed(l))  
View Answer

Answer: d  
Explanation: The built-in function reversed() can be used to reverse the elements of a list. This function accepts only an iterable as an argument. To print the output in the form of a list, we use: list(reversed(l)). The output will be: [4,3,2].

3. The output of the function:

float(' -12345\n')

(Note that the number of blank spaces before the number is 5)  
a) -12345.0 (5 blank spaces before the number)  
b) -12345.0  
c) Error  
d) -12345.000000000…. (infinite decimal places)  
View Answer

Answer: b  
Explanation: The function float() will remove all the blank spaces and covert the integer to a floating point number. Hence the output will be: -12345.0.

4. What is the output of the functions shown below?

ord(65)

ord(‘A’)

a) A  
65  
b) Error  
65  
c) A  
Error  
c) Error  
Error  
View Answer

Answer: b  
Explanation: The built-in function ord() is used to return the ASCII value of the alphabet passed to it as an argument. Hence the first function results in an error and the output of the second function is 65.

5. What is the output of the functions shown below?

float(‘-infinity’)

float(‘inf’)

a) –inf  
inf  
b) –infinity  
inf  
c) Error  
Error  
d) Error  
Junk value  
View Answer

Answer: a  
Explanation: The output of the first function will be –inf and that of the second function will be inf.

6. Which of the following functions will not result in an error when no arguments are passed to it?  
a) min()  
b) divmod()  
c) all()  
d) float()  
View Answer

Answer: d  
Explanation: The built-in functions min(), max(), divmod(), ord(), any(), all() etc throw an error when no arguments are passed to them. However there are some built-in functions like float(), complex() etc which do not throw an error when no arguments are passed to them. The output of float() is 0.0.

7. What is the output of the function shown below?

hex(15)

a) f  
b) 0xF  
c) 0Xf  
d) 0xf  
View Answer

Answer: d  
Explanation: The function hex() is used to convert the given argument into its hexadecimal representation, in lower case. Hence the output of the function hex(15) is 0xf.

8. Which of the following functions does not throw an error?  
a) ord()  
b) ord(‘ ‘)  
c) ord(”)  
d) ord(“”)  
View Answer

Answer: b  
Explanation: The function ord() accepts a character. Hence ord(), ord(”) and ord(“”) throw errors. However the function ord(‘ ‘) does not throw an error because in this case, we are actually passing a blank space as an argument. The output of ord(‘ ‘) is 32 (ASCII value corresponding to blank space).

9. What is the output of the function:

len(["hello",2, 4, 6])

a) 4  
b) 3  
c) Error  
d) 6  
View Answer

Answer: a  
Explanation: The function len() returns the length of the number of elements in the iterable. Therefore the output of the function shown above is 4.

10. What is the output of the function shown below?

oct(7)

oct(‘7’)

a) Error  
07  
b) 0o7  
Error  
c) 0o7  
Error  
d) 07  
0o7  
View Answer

Answer: c  
Explanation: The function oct() is used to convert its argument into octal form. This function does not accept strings. Hence the second function results in an error while the output of the first function is 0o7.

1. Which of the following is the use of function in python?  
a) Functions are reusable pieces of programs  
b) Functions don’t provide better modularity for your application  
c) you can’t also create your own functions  
d) All of the mentioned  
View Answer

Answer: a  
Explanation: Functions are reusable pieces of programs. They allow you to give a name to a block of statements, allowing you to run that block using the specified name anywhere in your program and any number of times.

2. Which keyword is use for function?  
a) Fun  
b) Define  
c) Def  
d) Function  
View Answer

Answer: c  
Explanation: None.

3. What is the output of the below program?

1. def sayHello():
2. print('Hello World!')
3. sayHello()
4. sayHello()

a) Hello World!  
Hello World!  
b) ‘Hello World!’  
‘Hello World!’  
c) Hello  
Hello  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: Functions are defined using the def keyword. After this keyword comes an identifier name for the function, followed by a pair of parentheses which may enclose some names of variables, and by the final colon that ends the line. Next follows the block of statements that are part of this function.

1. def sayHello():
2. print('Hello World!') # block belonging to the function
3. # End of function #
5. sayHello() # call the function
6. sayHello() # call the function again

4. What is the output of the below program?

1. def printMax(a, b):
2. if a > b:
3. print(a, 'is maximum')
4. elif a == b:
5. print(a, 'is equal to', b)
6. else:
7. print(b, 'is maximum')
8. printMax(3, 4)

a) 3  
b) 4  
c) 4 is maximum  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: Here, we define a function called printMax that uses two parameters called a and b. We find out the greater number using a simple if..else statement and then print the bigger number.

5. What is the output of the below program ?

1. x = 50
2. def func(x):
3. print('x is', x)
4. x = 2
5. print('Changed local x to', x)
6. func(x)
7. print('x is now', x)

a) x is now 50  
b) x is now 2  
c) x is now 100  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: The first time that we print the value of the name x with the first line in the function’s body, Python uses the value of the parameter declared in the main block, above the function definition.  
Next, we assign the value 2 to x. The name x is local to our function. So, when we change the value of x in the function, the x defined in the main block remains unaffected.  
With the last print function call, we display the value of x as defined in the main block, thereby confirming that it is actually unaffected by the local assignment within the previously called function.

6. What is the output of the below program?

1. x = 50
2. def func():
3. global x
4. print('x is', x)
5. x = 2
6. print('Changed global x to', x)
7. func()
8. print('Value of x is', x)

a) x is 50  
Changed global x to 2  
Value of x is 50  
b) x is 50  
Changed global x to 2  
Value of x is 2  
c) x is 50  
Changed global x to 50  
Value of x is 50  
d) None of the mentioned  
View Answer

Answer: b  
Explanation: The global statement is used to declare that x is a global variable – hence, when we assign a value to x inside the function, that change is reflected when we use the value of x in the main block.

7. What is the output of below program?

1. def say(message, times = 1):
2. print(message \* times)
3. say('Hello')
4. say('World', 5)

a) Hello  
WorldWorldWorldWorldWorld  
b) Hello  
World 5  
c) Hello  
World,World,World,World,World  
d) Hello  
HelloHelloHelloHelloHello  
View Answer

Answer: a  
Explanation: For some functions, you may want to make some parameters optional and use default values in case the user does not want to provide values for them. This is done with the help of default argument values. You can specify default argument values for parameters by appending to the parameter name in the function definition the assignment operator (=) followed by the default value.  
The function named say is used to print a string as many times as specified. If we don’t supply a value, then by default, the string is printed just once. We achieve this by specifying a default argument value of 1 to the parameter times.  
In the first usage of say, we supply only the string and it prints the string once. In the second usage of say, we supply both the string and an argument 5 stating that we want to say the string message 5 times.

8. What is the output of the below program?

1. def func(a, b=5, c=10):
2. print('a is', a, 'and b is', b, 'and c is', c)
4. func(3, 7)
5. func(25, c = 24)
6. func(c = 50, a = 100)

a) a is 7 and b is 3 and c is 10  
a is 25 and b is 5 and c is 24  
a is 5 and b is 100 and c is 50  
b) a is 3 and b is 7 and c is 10  
a is 5 and b is 25 and c is 24  
a is 50 and b is 100 and c is 5  
c) a is 3 and b is 7 and c is 10  
a is 25 and b is 5 and c is 24  
a is 100 and b is 5 and c is 50  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: If you have some functions with many parameters and you want to specify only some of them, then you can give values for such parameters by naming them – this is called keyword arguments – we use the name (keyword) instead of the position (which we have been using all along) to specify the arguments to the function.  
The function named func has one parameter without a default argument value, followed by two parameters with default argument values.

In the first usage, func(3, 7), the parameter a gets the value 3, the parameter b gets the value 7 and c gets the default value of 10.

In the second usage func(25, c=24), the variable a gets the value of 25 due to the position of the argument. Then, the parameter c gets the value of 24 due to naming i.e. keyword arguments. The variable b gets the default value of 5.

In the third usage func(c=50, a=100), we use keyword arguments for all specified values. Notice that we are specifying the value for parameter c before that for a even though a is defined before c in the function definition.

9. What is the output of below program?

1. def maximum(x, y):
2. if x > y:
3. return x
4. elif x == y:
5. return 'The numbers are equal'
6. else:
7. return y
9. print(maximum(2, 3))

a) 2  
b) 3  
c) The numbers are equal  
d) None of the mentioned  
View Answer

Answer: b  
Explanation: The maximum function returns the maximum of the parameters, in this case the numbers supplied to the function. It uses a simple if..else statement to find the greater value and then returns that value.

10. Which of the following is a features of DocString?  
a) Provide a convenient way of associating documentation with Python modules, functions, classes, and methods  
b) All functions should have a docstring  
c) Docstrings can be accessed by the \_\_doc\_\_ attribute on objects  
d) All of the mentioned  
View Answer

Answer: d  
Explanation: Python has a nifty feature called documentation strings, usually referred to by its shorter name docstrings. DocStrings are an important tool that you should make use of since it helps to document the program better and makes it easier to understand.

1. Which are the advantages of functions in python?  
a) Reducing duplication of code  
b) Decomposing complex problems into simpler pieces  
c) Improving clarity of the code  
d) All of the mentioned  
View Answer

Answer: d  
Explanation: None.

2. What are the two main types of functions?  
a) Custom function  
b) Built-in function & User defined function  
c) User function  
d) System function  
View Answer

Answer: b  
Explanation: Built-in functions and user defined ones. The built-in functions are part of the Python language. Examples are: dir(), len() or abs(). The user defined functions are functions created with the def keyword.

3. Where is function defined?  
a) Module  
b) Class  
c) Another function  
d) All of the mentioned  
View Answer

Answer: d  
Explanation: Functions can be defined inside a module, a class or another function.

4. What is called when a function is defined inside a class?  
a) Module  
b) Class  
c) Another function  
d) Method  
View Answer

Answer: d  
Explanation: None.

5. Which of the following is the use of id() function in python?  
a) Id returns the identity of the object  
b) Every object doesn’t have a unique id  
c) All of the mentioned  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: Each object in Python has a unique id. The id() function returns the object’s id.

6. Which of the following refers to mathematical function?  
a) sqrt  
b) rhombus  
c) add  
d) rhombus  
View Answer

Answer: a  
Explanation: Functions that are always available for usage, functions that are contained within external modules, which must be imported and functions defined by a programmer with the def keyword.  
Eg: math import sqrt  
The sqrt() function is imported from the math module.

7. What is the output of below program?

1. def cube(x):
2. return x \* x \* x
3. x = cube(3)
4. print x

a) 9  
b) 3  
c) 27  
d) 30  
View Answer

Answer: c  
Explanation: A function is created to do a specific task. Often there is a result from such a task. The return keyword is used to return values from a function. A function may or may not return a value. If a function does not have a return keyword, it will send a none value.

8. What is the output of the below program?

1. def C2F(c):
2. return c \* 9/5 + 32
3. print C2F(100)
4. print C2F(0)

a) 212  
32  
b) 314  
24  
c) 567  
98  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: The code shown above is used to convert a temperature in degree celsius to fahrenheit.

9. What is the output of the below program?

1. def power(x, y=2):
2. r = 1
3. for i in range(y):
4. r = r \* x
5. return r
6. print power(3)
7. print power(3, 3)

a) 212  
32  
b) 9  
27  
c) 567  
98  
d) None of the mentioned  
View Answer

Answer: b  
Explanation: The arguments in Python functions may have implicit values. An implicit value is used, if no value is provided. Here we created a power function. The function has one argument with an implicit value. We can call the function with one or two arguments.

10. What is the output of the below program?

1. def sum(\*args):
2. '''Function returns the sum
3. of all values'''
4. r = 0
5. for i in args:
6. r += i
7. return r
8. print sum.\_\_doc\_\_
9. print sum(1, 2, 3)
10. print sum(1, 2, 3, 4, 5)

a) 6  
15  
b) 6  
100  
c) 123  
12345  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: We use the \* operator to indicate, that the function will accept arbitrary number of arguments. The sum() function will return the sum of all arguments. The first string in the function body is called the function documentation string. It is used to document the function. The string must be in triple quotes.

1. Python supports the creation of anonymous functions at runtime, using a construct called \_\_\_\_\_\_\_\_\_\_  
a) Lambda  
b) pi  
c) anonymous  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: Python supports the creation of anonymous functions (i.e. functions that are not bound to a name) at runtime, using a construct called lambda. Lambda functions are restricted to a single expression. They can be used wherever normal functions can be used.

2. What is the output of this program?

1. y = 6
2. z = lambda x: x \* y
3. print z(8)

a) 48  
b) 14  
c) 64  
d) None of the mentioned  
View Answer

Answer: a  
Explanation: The lambda keyword creates an anonymous function. The x is a parameter, that is passed to the lambda function. The parameter is followed by a colon character. The code next to the colon is the expression that is executed, when the lambda function is called. The lambda function is assigned to the z variable.  
The lambda function is executed. The number 8 is passed to the anonymous function and it returns 48 as the result. Note that z is not a name for this function. It is only a variable to which the anonymous function was assigned.

3. What is the output of below program?

1. lamb = lambda x: x \*\* 3
2. print(lamb(5))

a) 15  
b) 555  
c) 125  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: None.

4. Does Lambda contains return statements?  
a) True  
b) False  
View Answer

Answer: b  
Explanation: lambda definition does not include a return statement. it always contains an expression which is returned. Also note that we can put a lambda definition anywhere a function is expected. We don’t have to assign it to a variable at all.

5. Lambda is a statement.  
a) True  
b) False  
View Answer

Answer: b  
Explanation: lambda is an anonymous function in Python. Hence this statement is false.

6. Lambda contains block of statements  
a) True  
b) False  
View Answer

Answer: b  
Explanation: None.

7. What is the output of below program?

1. def f(x, y, z): return x + y + z
2. f(2, 30, 400)

a) 432  
b) 24000  
c) 430  
d) No output  
View Answer

Answer: a  
Explanation: None.

8. What is the output of below program?

1. def writer():
2. title = 'Sir'
3. name = (lambda x:title + ' ' + x)
4. return name
6. who = writer()
7. who('Arthur')

a) Arthur Sir  
b) Sir Arthur  
c) Arthur  
d) None of the mentioned  
View Answer

Answer: b  
Explanation: None.

9. What is the output of this program?

1. L = [lambda x: x \*\* 2,
2. lambda x: x \*\* 3,
3. lambda x: x \*\* 4]
5. for f in L:
6. print(f(3))

a) 27  
81  
343  
b) 6  
9  
12  
c) 9  
27  
81  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: None.

10. What is the output of this program?

1. min = (lambda x, y: x if x < y else y)
2. min(101\*99, 102\*98)

a) 9997  
b) 9999  
c) 9996  
d) None of the mentioned  
View Answer

Answer: c  
Explanation: None.

1. What is a variable defined outside a function referred to as?  
a) A static variable  
b) A global variable  
c) A local variable  
d) An automatic variable  
View Answer

Answer: b  
Explanation: The value of a variable defined outside all function definitions is referred to as a global variable and can be used by multiple functions of the program.

2. What is a variable defined inside a function referred to as?  
a) A global variable  
b) A volatile variable  
c) A local variable  
d) An automatic variable  
View Answer

Answer: c  
Explanation: The variable inside a function is called as local variable and the variable definition is confined only to that function.

3. What is the output of the following code?

i=0

def change(i):

i=i+1

return i

change(1)

print(i)

a) 1  
b) Nothing is displayed  
c) 0  
d) An exception is thrown  
View Answer

Answer: c  
Explanation: Any change made in to an immutable data type in a function isn’t reflected outside the function.

4. What is the output of the following piece of code?

def a(b):

b = b + [5]

c = [1, 2, 3, 4]

a(c)

print(len(c))

a) 4  
b) 5  
c) 1  
d) An exception is thrown  
View Answer

Answer: b  
Explanation: Since a list is mutable, any change made in the list in the function is reflected outside the function.

5. What is the output of the following code?

a=10

b=20

def change():

global b

a=45

b=56

change()

print(a)

print(b)

a)10  
56  
b)45  
56  
c)10  
20  
d)Syntax Error  
View Answer

Answer: a  
Explanation: The statement “global b” allows the global value of b to be accessed and changed. Whereas the variable a is local and hence the change isn’t reflected outside the function.

6. What is the output of the following code?

def change(i = 1, j = 2):

i = i + j

j = j + 1

print(i, j)

change(j = 1, i = 2)

a) An exception is thrown because of conflicting values  
b) 1 2  
c) 3 3  
d) 3 2  
View Answer

Answer: d  
Explanation: The values given during function call is taken into consideration, that is, i=2 and j=1.

7. What is the output of the following code?

def change(one, \*two):

print(type(two))

change(1,2,3,4)

a) Integer  
b) Tuple  
c) Dictionary  
d) An exception is thrown  
View Answer

Answer: b  
Explanation: The parameter two is a variable parameter and consists of (2,3,4). Hence the data type is tuple.

8. If a function doesn’t have a return statement, which of the following does the function return?  
a) int  
b) null  
c) None  
d) An exception is thrown without the return statement  
View Answer

Answer: c  
Explanation: A function can exist without a return statement and returns None if the function doesn’t have a return statement.

9. What is the output of the following code?

def display(b, n):

while n > 0:

print(b,end="")

n=n-1

display('z',3)

a) zzz  
b) zz  
c) An exception is executed  
d) Infinite loop  
View Answer

Answer: a  
Explanation: The loop runs three times and ‘z’ is printed each time.

10. What is the output of the following piece of code?

def find(a, \*\*b):

print(type(b))

find('letters',A='1',B='2')

a) String  
b) Tuple  
c) Dictionary  
d) An exception is thrown  
View Answer

Answer: c  
Explanation: b combines the remaining parameters into a dictionary.

1. What is the type of each element in sys.argv?  
a) set  
b) list  
c) tuple  
d) string  
View Answer

Answer: d  
Explanation: It is a list of strings.

2. What is the length of sys.argv?  
a) number of arguments  
b) number of arguments + 1  
c) number of arguments – 1  
d) none of the mentioned  
View Answer

Answer: b  
Explanation: The first argument is the name of the program itself. Therefore the length of sys.argv is one more than the number arguments.

3. What is the output of the following code?

def foo(k):

k[0] = 1

q = [0]

foo(q)

print(q)

a) [0].  
b) [1].  
c) [1, 0].  
d) [0, 1].  
View Answer

Answer: b  
Explanation: Lists are passed by reference.

4. How are keyword arguments specified in the function heading?  
a) one star followed by a valid identifier  
b) one underscore followed by a valid identifier  
c) two stars followed by a valid identifier  
d) two underscores followed by a valid identifier  
View Answer

Answer: c  
Explanation: Refer documentation.

5. How many keyword arguments can be passed to a function in a single function call?  
a) zero  
b) one  
c) zero or more  
d) one or more  
View Answer

Answer: c  
Explanation: zero keyword arguments may be passed if all the arguments have default values.

6. What is the output of the following code?

def foo(fname, val):

print(fname(val))

foo(max, [1, 2, 3])

foo(min, [1, 2, 3])

a) 3 1  
b) 1 3  
c) error  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: It is possible to pass function names as arguments to other functions.

7. What is the output of the following code?

def foo():

return total + 1

total = 0

print(foo())

a) 0  
b) 1  
c) error  
d) none of the mentioned  
View Answer

Answer: b  
Explanation: It is possible to read the value of a global variable directly.

8. What is the output of the following code?

def foo():

total += 1

return total

total = 0

print(foo())

a) 0  
b) 1  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: It is not possible to change the value of a global variable without explicitly specifying it.

9. What is the output of the following code?

def foo(x):

x = ['def', 'abc']

return id(x)

q = ['abc', 'def']

print(id(q) == foo(q))

a) True  
b) False  
c) None  
d) Error  
View Answer

Answer: b  
Explanation: A new object is created in the function.

10. What is the output of the following code?

def foo(i, x=[]):

x.append(i)

return x

for i in range(3):

print(foo(i))

a) [0] [1] [2].  
b) [0] [0, 1] [0, 1, 2].  
c) [1] [2] [3].  
d) [1] [1, 2] [1, 2, 3].  
View Answer

Answer: b  
Explanation: When a list is a default value, the same list will be reused.

1. What is the output of the following code?

def foo(k):

k = [1]

q = [0]

foo(q)

print(q)

a) [0].  
b) [1].  
c) [1, 0].  
d) [0, 1].  
View Answer

Answer: a  
Explanation: A new list object is created in the function and the reference is lost. This can be checked by comparing the id of k before and after k = [1].

2. How are variable length arguments specified in the function heading?  
a) one star followed by a valid identifier  
b) one underscore followed by a valid identifier  
c) two stars followed by a valid identifier  
d) two underscores followed by a valid identifier  
View Answer

Answer: a  
Explanation: Refer documentation.

3. Which module in the python standard library parses options received from the command line?  
a) getopt  
b) os  
c) getarg  
d) main  
View Answer

Answer: a  
Explanation: getopt parses options received from the command line.

4. What is the type of sys.argv?  
a) set  
b) list  
c) tuple  
d) string  
View Answer

Answer: b  
Explanation: It is a list of elements.

5. What is the value stored in sys.argv[0]?  
a) null  
b) you cannot access it  
c) the program’s name  
d) the first argument  
View Answer

Answer: c  
Explanation: Refer documentation.

6. How are default arguments specified in the function heading?  
a) identifier followed by an equal to sign and the default value  
b) identifier followed by the default value within back-ticks (“)  
c) identifier followed by the default value within square brackets ([])  
d) identifier  
View Answer

Answer: a  
Explanation: Refer documentation.

7. How are required arguments specified in the function heading?  
a) identifier followed by an equal to sign and the default value  
b) identifier followed by the default value within back-ticks (“)  
c) identifier followed by the default value within square brackets ([])  
d) identifier  
View Answer

Answer: d  
Explanation: Refer documentation.

8. What is the output of the following code?

def foo(x):

x[0] = ['def']

x[1] = ['abc']

return id(x)

q = ['abc', 'def']

print(id(q) == foo(q))

a) True  
b) False  
c) None  
d) Error  
View Answer

Answer: a  
Explanation: The same object is modified in the function.

9. Where are the arguments received from the command line stored?  
a) sys.argv  
b) os.argv  
c) argv  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: Refer documentation.

10. What is the output of the following?

def foo(i, x=[]):

x.append(x.append(i))

return x

for i in range(3):

y = foo(i)

print(y)

a) [[[0]], [[[0]], [1]], [[[0]], [[[0]], [1]], [2]]].  
b) [[0], [[0], 1], [[0], [[0], 1], 2]].  
c) [0, None, 1, None, 2, None].  
d) [[[0]], [[[0]], [1]], [[[0]], [[[0]], [1]], [2]]].  
View Answer

Answer: c  
Explanation: append() returns None.

1. The output of the code shown below is:

def f1():

x=15

print(x)

x=12

f1()

a) Error  
b) 12  
c) 15  
d) 1512  
View Answer

Answer: c  
Explanation: In the code shown above, x=15 is a local variable whereas x=12 is a global variable. Preference is given to local variable over global variable. Hence the output of the code shown above is 15.

2. What is the output of the code shown below?

def f1():

x=100

print(x)

x=+1

f1()

a) Error  
b) 100  
c) 101  
d) 99  
View Answer

Answer: b  
Explanation: The variable x is a local variable. It is first printed and then modified. Hence the output of this code is 100.

3. What is the output of the code shown below?

def san(x):

print(x+1)

x=-2

x=4

san(12)

a) 13  
b) 10  
c) 2  
d) 5  
View Answer

Answer: a  
Explanation: The value passed to the function san() is 12. This value is incremented by one and printed. Hence the output of the code shown above is 13.

4. What is the output of the code shown?

def f1():

global x

x+=1

print(x)

x=12

print("x")

a) Error  
b) 13  
c) 13  
x  
d) x  
View Answer

Answer: d  
Explanation: In the code shown above, the variable ‘x’ is declared as global within the function. Hence the output is ‘x’. Had the variable ‘x’ been a local variable, the output would have been:  
13  
x

5. What is the output of the code shown below?

def f1(x):

global x

x+=1

print(x)

f1(15)

print("hello")

a) error  
b) hello  
c) 16  
d) 16  
hello  
View Answer

Answer: a  
Explanation: The code shown above will result in an error because ‘x’ is a global variable. Had it been a local variable, the output would be: 16  
hello

6. What is the output of the following code?

x=12

def f1(a,b=x):

print(a,b)

x=15

f1(4)

a) Error  
b) 12 4  
c) 4 12  
d) 4 15  
View Answer

Answer: c  
Explanation: At the time of leader processing, the value of ‘x’ is 12. It is not modified later. The value passed to the function f1 is 4. Hence the output of the code shown above is 4 12.

7. What is the output of the code shown?

def f():

global a

print(a)

a = "hello"

print(a)

a = "world"

f()

print(a)

a) hello  
hello  
world  
b) world  
world  
hello  
c) hello  
world  
world  
d) world  
hello  
world  
View Answer

Answer: b  
Explanation: Since the variable ‘a’ has been explicitly specified as a global variable, the value of a passed to the function is ‘world’. Hence the output of this code is: world  
world  
hello

8. What is the output of the code shown below?

def f1(a,b=[]):

b.append(a)

return b

print(f1(2,[3,4]))

a) [3,2,4]  
b) [2,3,4]  
c) Error  
d) [3,4,2]  
View Answer

Answer: d  
Explanation: In the code shown above, the integer 2 is appended to the list [3,4]. Hence the output of the code is [3,4,2]. Both the variables a and b are local variables.

9. What is the output of the code shown below?

def f(p, q, r):

global s

p = 10

q = 20

r = 30

s = 40

print(p,q,r,s)

p,q,r,s = 1,2,3,4

f(5,10,15)

a) 1 2 3 4  
b) 5 10 15 4  
c) 10 20 30 40  
d) 5 10 15 40  
View Answer

Answer: c  
Explanation: The above code shows a combination of local and global variables. The output of this code is: 10 20 30 40

10. What is the output of the code shown below?

def f(x):

print("outer")

def f1(a):

print("inner")

print(a,x)

f(3)

f1(1)

a) outer  
error  
b) inner  
error  
c) outer  
inner  
d) error  
View Answer

Answer: a  
Explanation: The error will be caused due to the statement f1(1) because the function is nested. If f1(1) had been called inside the function, the output would have been different and there would be no error.

11. The output of code shown below is:

x = 5

def f1():

global x

x = 4

def f2(a,b):

global x

return a+b+x

f1()

total = f2(1,2)

print(total)

a) Error  
b) 7  
c) 8  
d) 15  
View Answer

Answer: b  
Explanation: In the code shown above, the variable ‘x’ has been declared as a global variable under both the functions f1 and f2. The value returned is a+b+x = 1+2+4 = 7.

12. What is the output of the code shown below?

x=100

def f1():

global x

x=90

def f2():

global x

x=80

print(x)

a) 100  
b) 90  
c) 80  
d) Error  
View Answer

Answer: a  
Explanation: The output of the code shown above is 100. This is because the variable ‘x’ has been declared as global within the functions f1 and f2.

13. Read the code shown below carefully and point out the global variables:

y, z = 1, 2

def f():

global x

x = y+z

a) x  
b) y and z  
c) x, y and z  
d) Neither x, nor y, nor z  
View Answer

Answer: c  
Explanation: In the code shown above, x, y and z are global variables inside the function f. y and z are global because they are not assigned in the function. x is a global variable because it is explicitly specified so in the code. Hence, x, y and z are global variables.

1. Which of the following data structures is returned by the functions globals() and locals()?  
a) list  
b) set  
c) dictionary  
d) tuple  
View Answer

Answer: c  
Explanation: Both the functions, that is, globals() and locals() return value of the data structure dictionary.

2. What is the output of the code shown below?

x=1

def cg():

global x

x=x+1

cg()

x

a) 2  
b) 1  
c) 0  
d) Error  
View Answer

Answer: a  
Explanation: Since ‘x’ has been declared a global variable, it can be modified very easily within the function. Hence the output is 2.

3. On assigning a value to a variable inside a function, it automatically becomes a global variable. State whether true or false.  
a) True  
b) False  
View Answer

Answer: b  
Explanation: On assigning a value to a variable inside a function, t automatically becomes a local variable. Hence the above statement is false.

4. What is the output of the code shown below?

e="butter"

def f(a): print(a)+e

f("bitter")

a) error  
b) butter  
error  
c) bitter  
error  
d) bitterbutter  
View Answer

Answer: c  
Explanation: The output of the code shown above will be ‘bitter’, followed by an error. The error is because the operand ‘+’ is unsupported on the types used above.

5. What happens if a local variable exists with the same name as the global variable you want to access?  
a) Error  
b) The local variable is shadowed  
c) Undefined behavior  
d) The global variable is shadowed  
View Answer

Answer: d  
Explanation: If a local variable exists with the same name as the local variable that you want to access, then the global variable is shadowed. That is, preference is given to the local variable.

6. What is the output of the code shown below?

a=10

globals()['a']=25

print(a)

a) 10  
b) 25  
c) Junk value  
d) Error  
View Answer

Answer: b  
Explanation: In the code shown above, the value of ‘a’ can be changed by using globals() function. The dictionary returned is accessed using key of the variable ‘a’ and modified to 25.

7. What is the output of this code?

def f(): x=4

x=1

f()

x

a) Error  
b) 4  
c) Junk value  
d) 1  
View Answer

Answer: d  
Explanation: In the code shown above, when we call the function f, a new namespace is created. The assignment x=4 is performed in the local namespace and does not affect the global namespace. Hence the output is 1.

8. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ returns a dictionary of the module namespace.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ returns a dictionary of the current namespace.  
a) locals()  
globals()  
b) locals()  
locals()  
c) globals()  
locals()  
d) globals()  
globals()  
View Answer

Answer: c  
Explanation: The function globals() returns a dictionary of the module namespace, whereas the function locals() returns a dictionary of the current namespace.

1. Which is the most appropriate definition for recursion?  
a) A function that calls itself  
b) A function execution instance that calls another execution instance of the same function  
c) A class method that calls another class method  
d) An in-built method that is automatically called  
View Answer

Answer: b  
Explanation: The appropriate definition for a recursive function is a function execution instance that calls another execution instance of the same function either directly or indirectly.

2. Only problems that are recursively defined can be solved using recursion. True or False?  
a) True  
b) False  
View Answer

Answer: b  
Explanation: There are many other problems can also be solved using recursion.

3. Which of these is false about recursion?  
a) Recursive function can be replaced by a non-recursive function  
b) Recursive functions usually take more memory space than non-recursive function  
c) Recursive functions run faster than non-recursive function  
d) Recursion makes programs easier to understand  
View Answer

Answer: c  
Explanation: The speed of a program using recursion is slower than the speed of its non-recursive equivalent.

4. Fill in the line of code for calculating the factorial of a number.

def fact(num):

if num == 0:

return 1

else:

return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a) num\*fact(num-1)

b) (num-1)\*(num-2)

c) num\*(num-1)

d) fact(num)\*fact(num-1)  
View Answer

Answer: a  
Explanation: Suppose n=5 then, 5\*4\*3\*2\*1 is returned which is the factorial of 5.

5. What is the output of the following piece of code?

def test(i,j):

if(i==0):

return j

else:

return test(i-1,i+j)

print(test(4,7))

a) 13  
b) 7  
c) Infinite loop  
d) 17  
View Answer

Answer: a  
Explanation: The test(i-1,i+j) part of the function keeps calling the function until the base condition of the function is satisfied.

6. What is the output of the following code?

l=[]

def convert(b):

if(b==0):

return l

dig=b%2

l.append(dig)

convert(b//2)

convert(6)

l.reverse()

for i in l:

print(i,end="")

a) 011  
b) 110  
c) 3  
d) Infinite loop  
View Answer

Answer: b  
Explanation: The above code gives the binary equivalent of the number.

7. What is tail recursion?  
a) A recursive function that has two base cases  
b) A function where the recursive functions leads to an infinite loop  
c) A recursive function where the function doesn’t return anything and just prints the values  
d) A function where the recursive call is the last thing executed by the function  
View Answer

Answer: d  
Explanation: A recursive function is tail recursive when recursive call is executed by the function in the last.

8. Observe the following piece of code?

def a(n):

if n == 0:

return 0

else:

return n\*a(n - 1)

def b(n, tot):

if n == 0:

return tot

else:

return b(n-2, tot-2)

a) Both a() and b() aren’t tail recursive  
b) Both a() and b() are tail recursive  
c) b() is tail recursive but a() isn’t  
d) a() is tail recursive but b() isn’t  
View Answer

Answer: c  
Explanation: A recursive function is tail recursive when recursive call is executed by the function in the last.

9. Which of the following statements is false about recursion?  
a) Every recursive function must have a base case  
b) Infinite recursion can occur if the base case isn’t properly mentioned  
c) A recursive function makes the code easier to understand  
d) Every recursive function must have a return value  
View Answer

Answer: d  
Explanation: A recursive function needn’t have a return value.

10. What is the output of the following piece of code?

def fun(n):

if (n > 100):

return n - 5

return fun(fun(n+11));

print(fun(45))

a) 50  
b) 100  
c) 74  
d) Infinite loop  
View Answer

Answer: b  
Explanation: The fun(fun(n+11)) part of the code keeps executing until the value of n becomes greater than 100, after which n-5 is returned and printed.

11. Recursion and iteration are the same programming approach. True or False?  
a) True  
b) False  
View Answer

12. What happens if the base condition isn’t defined in recursive programs?  
a) Program gets into an infinite loop  
b) Program runs once  
c) Program runs n number of times where n is the argument given to the function  
d) An exception is thrown  
View Answer

Answer: a  
Explanation: The program will run until the system gets out of memory.

13. Which of these is not true about recursion?  
a) Making the code look clean  
b) A complex task can be broken into sub-problems  
c) Recursive calls take up less memory  
d) Sequence generation is easier than a nested iteration  
View Answer

Answer: c  
Explanation: Recursive calls take up a lot of memory and time as memory is taken up each time the function is called.

14. Which of these is not true about recursion?  
a) The logic behind recursion may be hard to follow  
b) Recursive functions are easy to debug  
c) Recursive calls take up a lot of memory  
d) Programs using recursion take longer time than their non-recursive equivalent  
View Answer

Answer: b  
Explanation: Recursive functions may be hard to debug as the logic behind recursion may be hard to follow.

15. What is the output of the following piece of code?

def a(n):

if n == 0:

return 0

elif n == 1:

return 1

else:

return a(n-1)+a(n-2)

for i in range(0,4):

print(a(i),end=" ")

a) 0 1 2 3  
b) An exception is thrown  
c) 0 1 1 2 3  
d) 0 1 1 2  
View Answer

Answer: d  
Explanation: The above piece of code prints the Fibonacci series.

1. Which type of copy is shown in this code?

l1=[[10, 20], [30, 40], [50, 60]]

ls=list(l1)

ls

[[10, 20], [30, 40], [50, 60]]

a) Shallow copy  
b) Deep copy  
c) memberwise  
d) All of the mentioned  
View Answer

Answer: a  
Explanation: The code shown above depicts shallow copy. For deep copy, the command given is: l2 = l1.copyy().

2. What is the output of the code shown below?

l=[2, 3, [4, 5]]

l2=l.copy()

l2[0]=88

l

l2

a) [88, 2, 3, [4, 5]]  
[88, 2, 3, [4, 5]]  
b) [2, 3, [4, 5]]  
[88, 2, 3, [4, 5]]  
c) [88, 2, 3, [4, 5]]  
[2, 3, [4, 5]]  
d) [2, 3, [4, 5]]  
[2, 3, [4, 5]]  
View Answer

Answer: b  
Explanation: The code shown above depicts deep copy. In deep copy, the base address of the objects is not copied. Hence the modification done on one list does not affect the other list.

3. In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the base address of the objects are copied.  
In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the base address of the objects are not copied.  
a) deep. shallow  
b) memberwise, shallow  
c) shallow, deep  
d) deep, memberwise  
View Answer

Answer: c  
Explanation: In shallow copy, the base address of the objects are copied.  
In deep copy, the base address of the objects are not copied.  
Note that memberwise copy is another name for shallow copy.

4. The nested list undergoes shallow copy even when the list as a whole undergoes deep copy. State whether this statement is true or false.  
a) True  
b) False  
View Answer

Answer: a  
Explanation: A nested list undergoes shallow copy even when the list as a whole undergoes deep copy. Hence, this statement is true.

5. The output of the code shown below and state the type of copy that is depicted:

l1=[2, 4, 6, 8]

l2=[1, 2, 3]

l1=l2

l2

a) [2, 4, 6, 8], shallow copy  
b) [2, 4, 6, 8], deep copy  
c) [1, 2, 3], shallow copy  
d) [1, 2, 3], deep copy  
View Answer

Answer: c  
Explanation: The code shown above depicts shallow copy and the output of the code is: [1, 2, 3].

6. What is the output of the codes shown below?

l1=[10, 20, 30]

l2=l1

id(l1)==id(l2)

l2=l1.copy()

id(l1)==id(l2)

a) False, False  
b) False, True  
c) True, True  
d) True, False  
View Answer

Answer: d  
Explanation: The first code shown above represents shallow copy. Hence the output of the expression id(l1)==id(l2) is True. The second code depicts deep copy. Hence the output of the expression id(l1)==id(l2) in the second case is False.

7. What is the output of the code shown below?

l1=[1, 2, 3, [4]]

l2=list(l1)

id(l1)==id(l2)

a) True  
b) False  
c) Error  
d) Address of l1  
View Answer

Answer: b  
Explanation: The code shown above shows a nested list. A nested list will undergo shallow copy when the list as a whole undergoes deep copy. Hence the output of this code is False.

8. What is the output of the code shown below?

l1=[10, 20, 30, [40]]

l2=copy.deepcopy(l1)

l1[3][0]=90

l1

l2

a) [10, 20, 30, [40]]  
[10, 20, 30, 90]  
b) Error  
c) [10, 20, 30 [90]]  
[10, 20, 30, [40]]  
d) [10, 20, 30, [40]]  
[10, 20, 30, [90]]  
View Answer

Answer: c  
Explanation: The code shown above depicts deep copy. Hence at the end of the code, l1=[10, 20, 30, [90]] and l2=[10, 20, 30, [40]].

9. Fill in the blanks:  
In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the modification done on one list affects the other list.  
In \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ copy, the modification done on one list does not affect the other list.  
a) shallow, deep  
b) memberwise, shallow  
c) deep, shallow  
d) deep, memberwise  
View Answer

Answer: a  
Explanation: In shallow copy, the modification done on one list affects the other list.  
In deep copy, the modification done on one list does not affect the other list.

10. What is the output of the code shown below?

l1=[1, 2, 3, (4)]

l2=l1.copy()

l2

l1

a) [1, 2, 3, (4)]  
[1, 2, 3, 4]  
b) [1, 2, 3, 4]  
[1, 2, 3, (4)]  
c) [1, 2, 3, 4]  
[1, 2, 3, 4]  
d) [1, 2, 3, (4)]  
[1, 2, 3, (4)]  
View Answer

Answer: c  
Explanation: In the code shown above, the list l1 is enclosed in a tuple. When we print this list, it is printed as [1, 2, 3, 4]. Note the absence of the tuple. The code shown depicts deep copy. Hence the output of this program is: l1=[1, 2, 3, 4] and l2=[1, 2, 3, 4].

11. What is the output of the piece of code given below?

def check(n):

if n < 2:

return n % 2 == 0

return check(n - 2)

print(check(11))

a) False  
b) True  
c) 1  
d) An exception is thrown  
View Answer

Answer: a  
Explanation: The above piece of code checks recursively whether a number is even or odd.

12. What is the base case in the MergeSort algorithm when it is solved recursively?  
a) n=0  
b) n=1  
c) A list of length one  
d) An empty list  
View Answer

Answer: c  
Explanation: MergeSort algorithm implements the recursive algorithm and when the recursive function receives a list of length 1 which is the base case, the list is returned.

13. What is the output of the following piece of code?

a = [1, 2, 3, 4, 5]

b = lambda x: (b (x[1:]) + x[:1] if x else [])

print(b (a))

a) 1 2 3 4 5.  
b) [5,4,3,2,1].  
c) [].  
d) Error, lambda functions can’t be called recursively.  
View Answer

Answer: c  
Explanation: The above piece of code appends the first element of the list to a reversed sublist and reverses the list using recursion

1. The output of the code shown below is:

odd=lambda x: bool(x%2)

numbers=[n for n in range(10)]

print(numbers)

n=list()

for i in numbers:

if odd(i):

continue

else:

break

a) [0, 2, 4, 6, 8, 10]  
b) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
c) [1, 3, 5, 7, 9]  
d) Error  
View Answer

Answer: b  
Explanation: The code shown above returns a new list containing whole numbers up to 10 (excluding 10). Hence the output of the code is: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

2. What is the output of the code shown below?

f=lambda x:bool(x%2)

print(f(20), f(21))

a) False True  
b) False False  
c) True True  
d) True False  
View Answer

Answer: a  
Explanation: The code shown above will return true if the given argument is an odd number, and false if the given argument is an even number. Since the arguments are 20 and 21 respectively, the output of this code is: False True.

3. What is the output of the code shown below?

import functools

l=[1,2,3,4]

print(functools.reduce(lambda x,y:x\*y,l))

a) Error  
b) 10  
c) 24  
d) No output  
View Answer

Answer: c  
Explanation: The code shown above returns the product of all the elements of the list. Hence the output is 1\*2\*3\*4 = 24.

4. What is the output of the code shown?

l=[1, -2, -3, 4, 5]

def f1(x):

return x<2

m1=filter(f1, l)

print(list(m1))

a) [1, 4, 5 ]  
b) Error  
c) [-2, -3]  
d) [1, -2, -3]  
View Answer

Answer: d  
Explanation: The code shown above returns only those elements from the list, which are less than 2. The functional programming tool used to achieve this operation is filter. Hence the output of the code is:[1, -2, -3].

5. What is the output of the code shown below?

l=[-2, 4]

m=map(lambda x:x\*2, l)

print(m)

a) [-4, 16]  
b) Address of m  
c) Error  
d) -4  
16  
View Answer

Answer: b  
Explanation: The code shown above returns the address of m. Had we used the statement: print(list(m)), the output would have been: [-4, 16].

6. What is the output of the following code?

l=[1, -2, -3, 4, 5]

def f1(x):

return x<-1

m1=map(f1, l)

print(list(m1))

a) [False, False, False, False, False]  
b) [False, True, True, False, False]  
c) [True, False, False, True, True]  
d) [True, True, True, True, True]  
View Answer

Answer: b  
Explanation: This code shown returns a list which contains True if the corresponding element of the list is less than -1, and false if the corresponding element is greater than -1. Hence the output of the code shown above: [False, True, True, False, False].

7. What is the output of the code shown?

l=[1, 2, 3, 4, 5]

m=map(lambda x:2\*\*x, l)

print(list(m))

a) [1, 4, 9, 16, 25 ]  
b) [2, 4, 8, 16, 32 ]  
c) [1, 0, 1, 0, 1]  
d) Error  
View Answer

Answer: b  
Explanation: The code shown above prints a list containing each element of the list as the power of two. That is, the output is: [2, 4, 8, 16, 32].

8. What is the output of the code shown?

import functools

l=[1, 2, 3, 4, 5]

m=functools.reduce(lambda x, y:x if x>y else y, l)

print(m)

a) Error  
b) Address of m  
c) 1  
d) 5  
View Answer

Answer: d  
Explanation: The code shown above can be used to find the maximum of the elements from the given list. In the above code, this operation is achieved by using the programming tool reduce. Hence the output of the code shown above is 5.

9. What is the output of the code shown below?

l=[n for n in range(5)]

f=lambda x:bool(x%2)

print(f(3), f(1))

for i in range(len(l)):

if f(l[i]):

del l[i]

print(i)

a) True True  
1  
2  
Error  
b) False False  
1  
2  
c) True False  
1  
2  
Error  
d) False True  
1  
2  
View Answer

Answer: a  
Explanation: The code shown above prints true if the value entered as an argument is odd, else false is printed. Hence the output: True True. The error is due to the list index being out of range.

10. What is the output of the code shown?

m=reduce(lambda x: x-3 in range(4, 10))

print(list(m))

a) [1, 2, 3, 4, 5, 6, 7]  
b) No output  
c) [1, 2, 3, 4, 5, 6]  
d) Error  
View Answer

Answer: b  
Explanation: The code shown above will result in an error. This is because e have not imported functools. Further, ‘reduce’, as such is not defined. We should use functools.reduce to remove the error.

11. Which of the following numbers will not be a part of the output list of the code shown below?

def sf(a):

return a%3!=0 and a%5!=0

m=filter(sf, range(1, 31))

print(list(m))

a) 1  
b) 29  
c) 6  
d) 10  
View Answer

Answer: d  
Explanation: The output list of the code shown above will not contain any element that is divisible by 3 or 5. Hence the number which is not present in the output list is 10. The output list: [1, 2, 4, 7, 8, 11, 13, 14, 16, 17, 19, 22, 23, 26, 28, 29]

12. The single line equivalent of the code shown below is:

l=[1, 2, 3, 4, 5]

def f1(x):

return x<0

m1=filter(f1, l)

print(list(m1))

a) filter(lambda x:x<0, l)  
b) filter(lambda x, y: x<0, l)  
c) filter(reduce x<0, l)  
d) reduce(x: x<0, l)  
View Answer

Answer: a  
Explanation: The code shown above returns a new list containing only those elements from list l, which are less than 0. Since there are no such elements in the list l, the output of this code is: [].The single line equivalent of this code is filter(lambda x:x<0, l).

13. What is the output of the line of code shown below?

list(map((lambda x:x^2), range(10)))

a) [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]  
b) Error  
c) [2, 3, 0, 1, 6, 7, 4, 5, 10, 11]  
d) No output  
View Answer

Answer: c  
Explanation: The line of code shown above returns a list of each number from 1 to 10, after an XOR operation is performed on each of these numbers with 2. Hence the output of this code is: [2, 3, 0, 1, 6, 7, 4, 5, 10, 11]

14. What is the output of the line of code shown below?

list(map((lambda x:x\*\*2), filter((lambda x:x%2==0), range(10))))

a) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
b) [0, 4, 16, 36, 64]  
c) Error  
d) No output  
View Answer

Answer: b  
Explanation: The output list will contain each number up to 10 raised to 2, except odd numbers, that is, 1, 3, 5, 9. Hence the output of the code is: [0, 4, 16, 36, 64].

15. The output of the two codes shown below is the same. State whether true or false.

[x\*\*2 for x in range(10)]

list(map((lambda x:x\*\*2), range(10)))

a) True  
b) False  
View Answer

Answer: a  
Explanation: Both of the codes shown above print each whole number up to 10, raised to the power 2. Hence the output of both of these codes is: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81].Therefore, the statement is true.

1. What is the output of the following?

elements = [0, 1, 2]

def incr(x):

return x+1

print(list(map(elements, incr)))

a) [1, 2, 3].  
b) [0, 1, 2].  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: The list should be the second parameter to the mapping function.

2. What is the output of the following?

elements = [0, 1, 2]

def incr(x):

return x+1

print(list(map(incr, elements)))

a) [1, 2, 3].  
b) [0, 1, 2].  
c) error  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: Each element of the list is incremented.

3. What is the output of the following?

x = ['ab', 'cd']

print(list(map(upper, x)))

a) [‘AB’, ‘CD’].  
b) [‘ab’, ‘cd’].  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: A NameError occurs because upper is a class method.

4. What is the output of the following?

def to\_upper(k):

return k.upper()

x = ['ab', 'cd']

print(list(map(upper, x)))

a) [‘AB’, ‘CD’].  
b) [‘ab’, ‘cd’].  
c) none of the mentioned  
d) error  
View Answer

Answer: d  
Explanation: A NameError occurs because upper is a class method.

5. What is the output of the following?

def to\_upper(k):

return k.upper()

x = ['ab', 'cd']

print(list(map(to\_upper, x)))

a) [‘AB’, ‘CD’].  
b) [‘ab’, ‘cd’].  
c) none of the mentioned  
d) error  
View Answer

Answer: a  
Explanation: Each element of the list is converted to uppercase.

6. What is the output of the following?

def to\_upper(k):

k.upper()

x = ['ab', 'cd']

print(list(map(to\_upper, x)))

a) [‘AB’, ‘CD’].  
b) [‘ab’, ‘cd’].  
c) none of the mentioned  
d) error  
View Answer

Answer: c  
Explanation: A list of Nones is printed as to\_upper() returns None.

7. What is the output of the following?

x = ['ab', 'cd']

print(map(len, x))

a) [‘ab’, ‘cd’].  
b) [2, 2].  
c) [‘2’, ‘2’].  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: A map object is generated by map(). We must convert this to a list to be able to print it in a human readable form.

8. What is the output of the following?

x = ['ab', 'cd']

print(list(map(len, x)))

a) [‘ab’, ‘cd’].  
b) [2, 2].  
c) [‘2’, ‘2’].  
d) none of the mentioned  
View Answer

Answer: b  
Explanation: The length of each string is 2.

9. What is the output of the following?

x = ['ab', 'cd']

print(len(map(list, x)))

a) [2, 2].  
b) 2  
c) 4  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: A TypeError occurs as map has no len().

10. What is the output of the following?

x = ['ab', 'cd']

print(len(list(map(list, x))))

a) 2  
b) 4  
c) error  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: The outer list has two lists in it. So it’s length is 2.

1. What is the output of the following?

x = ['ab', 'cd']

print(len(list(map(list, x))))))

a) 2  
b) 4  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: SyntaxError, unbalanced parenthesis.

2. What is the output of the following?

x = ['ab', 'cd']

print(list(map(list, x)))

a) [‘a’, ‘b’, ‘c’, ‘d’].  
b) [[‘ab’], [‘cd’]].  
c) [[‘a’, ‘b’], [‘c’, ‘d’]].  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: Each element of x is converted into a list.

3. What is the output of the following?

x = [12, 34]

print(len(list(map(len, x))))

a) 2  
b) 1  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: TypeError, int has no len().

4. What is the output of the following?

x = [12, 34]

print(len(list(map(int, x))))

a) 2  
b) 1  
c) error  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: list(map()) returns a list of two items in this example.

5. What is the output of the following?

x = [12, 34]

print(len(''.join(list(map(int, x)))))

a) 4  
b) 2  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: Cannot perform join on a list of ints.

6. What is the output of the following?

x = [12, 34]

print(len(''.join(list(map(str, x)))))

a) 4  
b) 5  
c) 6  
d) error  
View Answer

Answer: a  
Explanation: Each number is mapped into a string of length 2.

7. What is the output of the following?

x = [12, 34]

print(len(' '.join(list(map(int, x)))))

a) 4  
b) 5  
c) 6  
d) error  
View Answer

Answer: d  
Explanation: TypeError. Execute in shell to verify.

8. What is the output of the following?

x = [12.1, 34.0]

print(len(' '.join(list(map(str, x)))))

a) 6  
b) 8  
c) 9  
d) error  
View Answer

Answer: c  
Explanation: The floating point numbers are converted to strings and joined with a space between them.

9. What is the output of the following?

x = [12.1, 34.0]

print(' '.join(list(map(str, x))))

a) 12 1 34 0  
b) 12.1 34  
c) 121 340  
d) 12.1 34.0  
View Answer

Answer: d  
Explanation: str(ab.c) is ‘ab.c’.

10. What is the output of the following?

x = [[0], [1]]

print(len(' '.join(list(map(str, x)))))

a) 2  
b) 3  
c) 7  
d) 8  
View Answer

Answer: c  
Explanation: map() is applied to the elements of the outer loop.

1. What is the output of the following?

x = [[0], [1]]

print((' '.join(list(map(str, x)))))

a) (‘[0] [1]’,)  
b) (’01’,)  
c) [0] [1].  
d) 01  
View Answer

Answer: c  
Explanation: (element) is the same as element. It is not a tuple with one item.

2. What is the output of the following?

x = [[0], [1]]

print((' '.join(list(map(str, x))),))

a) (‘[0] [1]’,)  
b) (’01’)  
c) [0] [1].  
d) 01  
View Answer

Answer: a  
Explanation: (element,) is not the same as element. It is a tuple with one item.

3. What is the output of the following?

x = [34, 56]

print((''.join(list(map(str, x))),))

a) 3456  
b) (3456)  
c) (‘3456’)  
d) (‘3456’,)  
View Answer

Answer: d  
Explanation: We have created a tuple with one string in it.

4. What is the output of the following?

x = [34, 56]

print((''.join(list(map(str, x)))),)

a) 3456  
b) (3456)  
c) (‘3456’)  
d) (‘3456’,)  
View Answer

Answer: a  
Explanation: We have just created a string.

5. What is the output of the following?

x = [34, 56]

print(len(map(str, x)))

a) [34, 56].  
b) [’34’, ’56’].  
c) 34 56  
d) error  
View Answer

Answer: d  
Explanation: TypeError, map has no len.

6. What is the output of the following?

x = 'abcd'

print(list(map(list, x)))

a) [‘a’, ‘b’, ‘c’, ‘d’].  
b) [‘abcd’].  
c) [[‘a’], [‘b’], [‘c’], [‘d’]].  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: list() is performed on each character in x.

7. What is the output of the following?

x = abcd

print(list(map(list, x)))

a) [‘a’, ‘b’, ‘c’, ‘d’].  
b) [‘abcd’].  
c) [[‘a’], [‘b’], [‘c’], [‘d’]].  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: NameError, we have not defined abcd.

8. What is the output of the following?

x = 1234

print(list(map(list, x)))

a) [1, 2, 3, 4].  
b) [1234].  
c) [[1], [2], [3], [4]].  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: TypeError, int is not iterable.

9. What is the output of the following?

x = 1234

print(list(map(list, [x])))

a) [1, 2, 3, 4].  
b) [1234].  
c) [[1], [2], [3], [4]].  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: TypeError, int is not iterable.

10. What is the output of the following?

x = 'abcd'

print(list(map([], x)))

a) [‘a’, ‘b’, ‘c’, ‘d’].  
b) [‘abcd’].  
c) [[‘a’], [‘b’], [‘c’], [‘d’]].  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: TypeError, list object is not callable.

11. Is Python code compiled or interpreted?  
a) Python code is only compiled  
b) Python code is both compiled and interpreted  
c) Python code is only interpreted  
d) Python code is neither compiled nor interpreted  
View Answer

Answer: b  
Explanation: Many languages have been implemented using both compilers and interpreters, including C, Pascal, and Python.

12. Which of these is the definition for packages in Python?  
a) A folder of python modules  
b) A set of programs making use of Python modules  
c) A set of main modules  
d) A number of files containing Python definitions and statements  
View Answer

Answer: a  
Explanation: A folder of python programs is called as a package of modules.

13. Which of these is false about a package?  
a) A package can have subfolders and modules  
b) Each import package need not introduce a namespace  
c) import folder.subfolder.mod1 imports packages  
d) from folder.subfolder.mod1 import objects imports packages  
View Answer

Answer: b  
Explanation: Packages provide a way of structuring Python’s namespace. Each import package introduces a namespace.

1. Which of these definitions correctly describes a module?  
a) Denoted by triple quotes for providing the specification of certain program elements  
b) Design and implementation of specific functionality to be incorporated into a program  
c) Defines the specification of how it is to be used  
d) Any program that reuses code  
View Answer

Answer: b  
Explanation: The term “module” refers to the implementation of specific functionality to be incorporated into a program.

2. Which of the following is not an advantage of using modules?  
a) Provides a means of reuse of program code  
b) Provides a means of diving up tasks  
c) Provides a means of reducing the size of the program  
d) Provides a means of testing individual parts of the program  
View Answer

Answer: c  
Explanation: The total size of the program remains the same regardless of whether modules are used or not. Modules simply divide the program.

3. Program code making use of a given module is called a \_\_\_\_\_\_ of the module.  
a) Client  
b) Docstring  
c) Interface  
d) Modularity  
View Answer

Answer: a  
Explanation: Program code making use of a given module is called the client of the module. There may be multiple clients for a module.

4. \_\_\_\_\_\_ is a string literal denoted by triple quotes for providing the specifications of certain program elements.  
a) Interface  
b) Modularity  
c) Client  
d) Docstring  
View Answer

Answer: d  
Explanation: Docstring used for providing the specifications of program elements.

5. Which of the following is true about top-down design process?  
a) The details of a program design are addressed before the overall design  
b) Only the details of the program are addressed  
c) The overall design of the program is addressed before the details  
d) Only the design of the program is addressed  
View Answer

Answer: c  
Explanation: Top-down design is an approach for deriving a modular design in which the overall design.

6. In top-down design every module is broken into same number of submodules? True or False?  
a) True  
b) False  
View Answer

Answer: b  
Explanation: In top-down design every module can even be broken down into different number of submodules.

7. All modular designs are because of a top-down design process? True or False?  
a) True  
b) False  
View Answer

Answer: b  
Explanation: The details of the program can be addressed before the overall design too. Hence, all modular designs are not because of a top-down design process.

8. What is the output of the following piece of code?

#mod1

def change(a):

b=[x\*2 for x in a]

print(b)

#mod2

def change(a):

b=[x\*x for x in a]

print(b)

from mod1 import change

from mod2 import change

#main

s=[1,2,3]

change(s)

a) [2,4,6].  
b) [1,4,9].  
c) [2,4,6].  
[1,4,9].  
d) There is a name clash.  
View Answer

Answer: d  
Explanation: A name clash is when two different entities with the same identifier become part of the same scope. Since both the modules have the same function name, there is a name clash.

9. Which of the following isn’t true about main modules?  
a) When a python file is directly executed, it is considered main module of a program  
b) Main modules may import any number of modules  
c) Special name given to main modules is: \_\_main\_\_  
d) Other main modules can import main modules  
View Answer

Answer: d  
Explanation: Main modules are not meant to be imported into other modules.

10. Which of the following is not a valid namespace?  
a) Global namespace  
b) Public namespace  
c) Built-in namespace  
d) Local namespace  
View Answer

Answer: b  
Explanation: During a Python program’s execution, there are as many as three namespaces – built-in namespace, global namespace and local namespace.

11. Which of the following is false about “import modulename” form of import?  
a) The namespace of imported module becomes part of importing module  
b) This form of import prevents name clash  
c) The namespace of imported module becomes available to importing module  
d) The identifiers in module are accessed as: modulename.identifier  
View Answer

Answer: a  
Explanation: In the “import modulename” form of import, the namespace of imported module becomes available to, but not part of, the importing module.

12. Which of the following is false about “from-import” form of import?  
a) The syntax is: from modulename import identifier  
b) This form of import prevents name clash  
c) The namespace of imported module becomes part of importing module  
d) The identifiers in module are accessed directly as: identifier  
View Answer

Answer: b  
Explanation: In the “from-import” form of import, there may be name clashes because names of the imported identifiers aren’t specified along with the module name.

13. Which of the statements about modules is false?  
a) In the “from-import” form of import, identifiers beginning with two underscores are private and aren’t imported  
b) dir() built-in function monitors the items in the namespace of the main module  
c) In the “from-import” form of import, all identifiers regardless of whether they are private or public are imported  
d) When a module is loaded, a compiled version of the module with file extension .pyc is automatically produced  
View Answer

Answer: c  
Explanation: In the “from-import” form of import, identifiers beginning with two underscores are private and aren’t imported.

14. What is the output of the following piece of code?

from math import factorial

print(math.factorial(5))

a) 120  
b) Nothing is printed  
c) Error, method factorial doesn’t exist in math module  
d) Error, the statement should be: print(factorial(5))  
View Answer

Answer: d  
Explanation: In the “from-import” form of import, the imported identifiers (in this case factorial()) aren’t specified along with the module name.

15. What is the order of namespaces in which Python looks for an identifier?  
a) Python first searches the global namespace, then the local namespace and finally the built-in namespace  
b) Python first searches the local namespace, then the global namespace and finally the built-in namespace  
c) Python first searches the built-in namespace, then the global namespace and finally the local namespace  
d) Python first searches the built-in namespace, then the local namespace and finally the global namespace  
View Answer

Answer: b  
Explanation: Python first searches for the local, then the global and finally the built-in namespace.

1. What is returned by math.ceil(3.4)?  
a) 3  
b) 4  
c) 4.0  
d) 3.0  
View Answer

Answer: b  
Explanation: The ceil function returns the smallest integer that is bigger than or equal to the number itself.

2. What is the value returned by math.floor(3.4)?  
a) 3  
b) 4  
c) 4.0  
d) 3.0  
View Answer

Answer: a  
Explanation: The floor function returns the biggest number that is smaller than or equal to the number itself.

3. What is the output of print(math.copysign(3, -1))?  
a) 1  
b) 1.0  
c) -3  
d) -3.0  
View Answer

Answer: d  
Explanation: The copysign function returns a float whose absolute value is that of the first argument and the sign is that of the second argument.

4. What is displayed on executing print(math.fabs(-3.4))?  
a) -3.4  
b) 3.4  
c) 3  
d) -3  
View Answer

Answer: b  
Explanation: A negative floating point number is returned as a positive floating point number.

5. Is the output of the function abs() the same as that of the function math.fabs()?  
a) sometimes  
b) always  
c) never  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: math.fabs() always returns a float and does not work with complex numbers whereas the return type of abs() is determined by the type of value that is passed to it.

6. What is the value returned by math.fact(6)?  
a) 720  
b) 6  
c) [1, 2, 3, 6].  
d) error  
View Answer

Answer: d  
Explanation: NameError, fact() is not defined.

7. What is the value of x if x = math.factorial(0)?  
a) 0  
b) 1  
c) error  
d) none of the mentioned  
View Answer

Answer: b  
Explanation: Factorial of 0 is 1.

8. What is math.factorial(4.0)?  
a) 24  
b) 1  
c) error  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: The factorial of 4 is returned.

9. What is the output of print(math.factorial(4.5))?  
a) 24  
b) 120  
c) error  
d) 24.0  
View Answer

Answer: c  
Explanation: Factorial is only defined for non-negative integers.

10. What is math.floor(0o10)?  
a) 8  
b) 10  
c) 0  
d) 9  
View Answer

Answer: a  
Explanation: 0o10 is 8 and floor(8) is 8.

1. What does the function math.frexp(x) return?  
a) a tuple containing of the mantissa and the exponent of x  
b) a list containing of the mantissa and the exponent of x  
c) a tuple containing of the mantissa of x  
d) a list containing of the exponent of x  
View Answer

Answer: a  
Explanation: It returns a tuple with two elements. The first element is the mantissa and the second element is the exponent.

2. What is the result of math.fsum([.1 for i in range(20)])?  
a) 2.0  
b) 20  
c) 2  
d) 2.0000000000000004  
View Answer

Answer: a  
Explanation: The function fsum returns an accurate floating point sum of the elements of its argument.

3. What is the result of sum([.1 for i in range(20)])?  
a) 2.0  
b) 20  
c) 2  
d) 2.0000000000000004  
View Answer

Answer: d  
Explanation: There is some loss of accuracy when we use sum with floating point numbers. Hence the function fsum is preferable.

4. What is returned by math.isfinite(float(‘inf’))?  
a) True  
b) False  
c) None  
d) error  
View Answer

Answer: b  
Explanation: float(‘inf’) is not a finite number.

5. What is returned by math.isfinite(float(‘nan’))?  
a) True  
b) False  
c) None  
d) error  
View Answer

Answer: b  
Explanation: float(‘nan’) is not a finite number.

6. What is x if x = math.isfinite(float(‘0.0’))?  
a) True  
b) False  
c) None  
d) error  
View Answer

Answer: a  
Explanation: float(‘0.0’) is a finite number.

7. What is the result of the following?

>>> -float('inf') + float('inf')

a) inf  
b) nan  
c) 0  
d) 0.0  
View Answer

Answer: b  
Explanation: The result of float(‘inf’)-float(‘inf’) is undefined.

8. What is the output of the following?

print(math.isinf(float('-inf')))

a) error, the minus sign shouldn’t havve been inside the brackets  
b) error, there is no function called isinf  
c) True  
d) False  
View Answer

Answer: c  
Explanation: -float(‘inf’) is the same as float(‘-inf’).

9. What is the value of x if x = math.ldexp(0.5, 1)?  
a) 1  
b) 2.0  
c) 0.5  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: The value returned by ldexp(x, y) is x \* (2 \*\* y). In the current case x is 1.0.

10. What is returned by math.modf(1.0)?  
a) (0.0, 1.0)  
b) (1.0, 0.0)  
c) (0.5, 1)  
d) (0.5, 1.0)  
View Answer

Answer: a  
Explanation: The first element is the fractional part and the second element is the integral part of the argument.

1. What is the result of math.trunc(3.1)?  
a) 3.0  
b) 3  
c) 0.1  
d) 1  
View Answer

Answer: b  
Explanation: The integral part of the floating point number is returned.

2. What is the output of print(math.trunc(‘3.1’))?  
a) 3  
b) 3.0  
c) error  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: TypeError, a string does not have \_\_trunc\_\_ method.

3. Which of the following is the same as math.exp(p)?  
a) e \*\* p  
b) math.e \*\* p  
c) p \*\* e  
d) p \*\* math.e  
View Answer

Answer: b  
Explanation: math.e is the constant defined in the math module.

4. What is returned by math.expm1(p)?  
a) (math.e \*\* p) – 1  
b) math.e \*\* (p – 1)  
c) error  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: One is subtracted from the result of math.exp(p) and returned.

5. What is the default base used when math.log(x) is found?  
a) e  
b) 10  
c) 2  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: The natural log of x is returned by default.

6. Which of the following aren’t defined in the math module?  
a) log2()  
b) log10()  
c) logx()  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: log2() and log10() are defined in the math module.

7. What is returned by int(math.pow(3, 2))?  
a) 6  
b) 9  
c) error, third argument required  
d) error, too many arguments  
View Answer

Answer: b  
Explanation: math.pow(a, b) returns a \*\* b.

8. What is output of print(math.pow(3, 2))?  
a) 9  
b) 9.0  
c) None  
d) none of the mentioned  
View Answer

Answer: b  
Explanation: math.pow() returns a floating point number.

9. What is the value of x if x = math.sqrt(4)?  
a) 2  
b) 2.0  
c) (2, -2)  
d) (2.0, -2.0)  
View Answer

Answer: b  
Explanation: The function returns one floating point number.

10. What does math.sqrt(X, Y) do?  
a) calculate the Xth root of Y  
b) calculate the Yth root of X  
c) error  
d) return a tuple with the square root of X and Y  
View Answer

Answer: c  
Explanation: The function takes only one argument.

1. The output of the snippet of code shown below is:

import datetime

d=datetime.date(2016,7,24)

print(d)

a) Error  
b) 2017-07-24  
c) 2017-7-24  
d) 24-7-2017  
View Answer

Answer: b  
Explanation: In the snippet of code shown above, we are simply printing the date entered by us. We enter the date in the format: yyyy,m,dd. The date is then printed in the format: yyyy-mm-dd. Hence the output is: 2017-07-24.

2. What is the output of the snippet of code shown below?

import datetime

d=datetime.date(2017,06,18)

print(d)

a) Error  
b) 2017-06-18  
c) 18-06-2017  
d) 06-18-2017  
View Answer

Answer: a  
Explanation: The code shown above will result in an error because of the format of the date entered. Had the date been entered as: d=datetime.date(2017,6,18), no error would have been thrown.

3. What is the output of the code shown below if the system date is 18th August, 2016?

tday=datetime.date.today()

print(tday.month())

a) August  
b) Aug  
c) 08  
d) 8  
View Answer

Answer: d  
Explanation: The code shown above prints the month number from the system date. Therefor the output will be 8 if the system date is 18th August, 2016.

4. What is the output of the code shown if the system date is 18th June, 2017 (Sunday)?

import datetime

tday=datetime.date.today()

print(tday)

a) 18-06-2017  
b) 06-18-2017  
c) 2017-06-18  
d) Error  
View Answer

Answer: c  
Explanation: The code shown above prints the system date in the format yyyy-mm-dd. Hence the output of this code is: 2017-06-18.

5. What is the output of the code shown below if the system date is 18th June, 2017 (Sunday)?

tday=datetime.date.today()

print(tday.weekday())

a) 6  
b) 1  
c) 0  
d) 7  
View Answer

Answer: a  
Explanation: The code shown above prints an integer depending on which day of the week it is. Monday-0, Tuesday-1, Wednesday-2, Thursday-3, Friday-4, Saturday-5, Sunday-6. Hence the output is 6 in the case shown above.

6. What is the output of the following code if the system date is 21st June, 2017 (Wednesday)?

tday=datetime.date.today()

print(tday.isoweekday())

a) Wed  
b) Wednesday  
c) 2  
d) 3  
View Answer

Answer: d  
Explanation: This code prints an integer depending on which day of the week it is. Monday-1, Tuesday-2, Wednesday-3, Thursday-4, Friday-5, Saturday-6, Sunday-7. Hence the output of the code shown above is 3.

7. Point out the error (if any) in the code shown below if the system date is 18th June, 2017?

tday=datetime.date.today()

bday=datetime.date(2017,9,18)

till\_bday=bday-tday

print(till\_bday)

a) 3 months, 0:00:00  
b) 90 days, 0:00:00  
c) 3 months 2 days, 0:00:00  
d) 92 days, 0:00:00  
View Answer

Answer: d  
Explanation: The code shown above can be used to find the number of days between two given dates. The output of the code shown above will thus be 92.

8. The value returned when we use the function isoweekday() is \_\_\_\_\_\_ and that for the function weekday() is \_\_\_\_\_\_\_\_ if the system date is 19th June, 2017 (Monday).  
a) 0,0  
b) 0,1  
c) 1,0  
d) 1,1  
View Answer

Answer: c  
Explanation: The value returned when we use the function isoweekday() is 1 and that for the function weekday() is 0 if the system date is 19th June, 2017 (Monday).

9. Which of the following will throw an error if used after the code shown below?

tday=datetime.date.today()

bday=datetime.date(2017,9,18)

t\_day=bday-tday

a) print(t\_day.seconds)  
b) print(t\_day.months)  
c) print(t\_day.max)  
d) print(t\_day.resolution)  
View Answer

Answer: b  
Explanation: The statement: print(t\_day.months) will throw an error because there is no function such as t\_day.months, whereas t\_day.seconds, t\_day.max and t\_day.resolution are valid, provided that t\_day is defined.

10. What is the output of the code shown below if the system date is: 6/19/2017

tday=datetime.date.today()

tdelta=datetime.timedelta(days=10)

print(tday+tdelta)

a) 2017-16-19  
b) 2017-06-9  
c) 2017-06-29  
d) Error  
View Answer

Answer: c  
Explanation: The code shown above will add the specified number of days to the current date and print the new date. On adding ten days to 6/19/2017, we get 6/29/2017. Hence the output is: 2017-06-29.

1. The output of both of the print statements is the same. State whether true or false.

import datetime

dt\_1 = datetime.datetime.today()

dt\_2 = datetime.datetime.now()

print(dt\_1)

print(dt\_2)

a) True  
b) False  
View Answer

Answer: b  
Explanation: The output of the two print statements is not the same because of the difference in time between the execution of the two print statements. There is a difference in the order of milli-seconds between the two statements and this is reflected in the output.

2. Which of the following functions can be used to find the coordinated universal time, assuming that the datetime module has already been imported?  
a) datetime.utc()  
b) datetime.datetime.utc()  
c) datetime.utcnow()  
d) datetime.datetime.utcnow()  
View Answer

Answer: d  
Explanation: The function datetime.datetime.utcnow() can be used to find the UTC (Coordinated Universal Time), assuming that the datetime module has already been imported. The other function s shown above are invalid.

3. What is the output of the code shown below?

import time

time.time()

a) The number of hours passed since 1st January, 1970  
b) The number of days passed since 1st January, 1970  
c) The number of seconds passed since 1st January, 1970  
d) The number of minutes passed since 1st January, 1970  
View Answer

Answer: c  
Explanation: The code shown above will return the number of seconds passed since 1st January, 1970.

4. What is the output of the following code, if the time module has already been imported?

def num(m):

t1 = time.time()

for i in range(0,m):

print(i)

t2 = time.time()

print(str(t2-t1))

num(3)

a) 1  
2  
3  
The time taken for the execution of the code  
b) 3  
The time taken for the execution of the code  
c) 1  
2  
3  
UTC time  
d) 3  
UTC time  
View Answer

Answer: a  
Explanation: The code shown above will return the numbers 1, 2, 3, followed by the time taken in the execution of the code. Hence option (a) shows the output correctly.

5. The output of the code shown below:

import time

time.asctime()

a) Current date only  
b) UTC time  
c) Current date and time  
d) Current time only  
View Answer

Answer: c  
Explanation: The function time.asctime(), present if the time module can be used to return the current date and time. It can also accept a parameter and return the date and time in a particular format. However in the above code, since we have not passed any parameters in the above code, the current date and time is returned.

6. What is the output of the code shown below?

import time

t=(2010, 9, 20, 8, 15, 12, 6)

time.asctime(t)

a) ‘20 Sep 2010 8:15:12 Sun’  
b) ‘2010 20 Sept 08:15:12 Sun’  
c) ‘Sun Sept 20 8:15:12 2010’  
d) Error  
View Answer

Answer: d  
Explanation: The code shown above results in an error because this function accepts exactly 9 arguments (including day of the year and DST), but only 7 are given. Hence an error is thrown.

7. What is the output of the code shown below?

import time

t=(2010, 9, 20, 8, 45, 12, 6, 0, 0)

time.asctime(t)

a) ‘Sep 20 2010 08:45:12 Sun’  
b) ‘Sun Sep 20 08:45:12 2010’  
c) ’20 Sep 08:45:12 Sun 2010’  
d) ‘2010 20 Sep 08:45:12 Sun’  
View Answer

Answer: b  
Explanation: The code shown above returns the given date and time in a particular format. Hence the output of the code shown above will be: ‘Sun Sep 20 08:45:12 2010’.

8. The sleep function (under the time module) is used to:  
a) Pause the code for the specified number of seconds  
b) Return the specified number of seconds, in terms of milli-seconds  
c) Stop the execution of the code  
d) Return the output of the code had it been executed earlier by the specified number of seconds  
View Answer

Answer: a  
Explanation: The sleep function (under the time module) is used to pause the code for the specified number of seconds. The number of seconds is taken as an argument by this function.

9. The output of the code shown will be:

import time

for i in range(0,5):

print(i)

time.sleep(2)

a) After an interval of 2 seconds, the numbers 1, 2, 3, 4, 5 are printed all together  
b) After an interval of 2 seconds, the numbers 0, 1, 2, 3, 4 are printed all together  
c) Prints the numbers 1, 2, 3, 4, 5 at an interval of 2 seconds between each number  
d) Prints the numbers 0, 1, 2, 3, 4 at an interval of 2 seconds between each number  
View Answer

Answer: d  
Explanation: The output of the code shown above will be the numbers 0, 1, 2, 3, 4 at an interval of 2 seconds each.

10. The output of the code shown below is  
time.struct\_time(tm\_year=2017, tm\_mon=6, tm\_mday=25, tm\_hour=18, tm\_min=26, tm\_sec=6, tm\_wday=6, tm\_yday=176, tm\_isdst=0)  
Code:

import time

t=time.localtime()

print(t)

To extract only the year from this, we can use the function:  
a) t[1]  
b) tm\_year  
c) t[0]  
d) t\_year  
View Answer

Answer: c  
Explanation: To extract the year from the code shown above, we use the command t[0]. The command t[1] will return the month number (6 in the above case). The commands tm\_year and t\_year will result in errors.

11. State whether true or false.

s = time.time()

t= time.time()

s == t

a) True  
b) False  
View Answer

Answer: b  
Explanation: The variables ‘s’ and ‘t’ will not be equal due to the slight difference in the time of their execution. Hence the output of this code will be: False.

1. To include the use of functions which are present in the random library, we must use the option:  
a) import random  
b) random.h  
c) import.random  
d) random.random  
View Answer

Answer: a  
Explanation: The command import random is used to import the random module, which enables us to use the functions which are present in the random library.

2. The output of the following snippet of code is either 1 or 2. State whether this statement is true or false.

import random

random.randint(1,2)

a) True  
b) False  
View Answer

Answer: a  
Explanation: The function random.randint(a,b) helps us to generate an integer between ‘a’ and ‘b’, including ‘a’ and ‘b’. In this case, since there are no integers between 1 and 2 , the output will necessarily be either 1 or 2’.

3. What is the output of the code shown below?

import random

random.choice(2,3,4)

a) An integer other than 2, 3 and 4  
b) Either 2, 3 or 4  
c) Error  
d) 3 only  
View Answer

Answer: c  
Explanation: The code shown above displays the incorrect syntax of the function random.choice(). This functions takes its numeric parameter in the form of a list. Hence the correct syntax world be: random.choice([2,3,4]).

4. What is the output of the code shown below?

import random

random.choice([10.4, 56.99, 76])

a) Error  
b) Either 10.4, 56.99 or 76  
c) Any number other than 10.4, 56.99 and 76  
c) 56.99 only  
View Answer

Answer: b  
Explanation: The function random.choice(a,b,c,d) returns a random number which is selected from a, b, c and d. The output can be either a, b, c or d. Hence the output of the snippet of code shown above can be either 10.4, 56.99 or 76.

5. What is the output of the function shown below (random module has already been imported)?

random.choice('sun')

a) sun  
b) u  
c) either s, u or n  
d) error  
View Answer

Answer: c  
Explanation: The above function works with alphabets just as it does with numbers. The output of thes expression will be either s, u or n.

6. What is the output of the following funtion, assuming that the random module has already been imported?

random.uniform(3,4)

a) Error  
b) Either 3 or 4  
c) Any integer other than 3 and 4  
d) Any decimal value between 3 and 4  
View Answer

Answer: d  
Explanation: This question depicts the basic difference between the functions random.randint(a, b) and random.uniform(a, b). While random.randint(a,b) generates an integer between ‘a’ and ‘b’, including ‘a’ and ‘b’, the function random.uniform(a,b) generates a decimal value between ‘a’ and ‘b’.

7. What is the output of the function shown below if the random module has already been imported?

random.randint(3.5,7)

a) Error  
b) Any integer between 3.5 and 7, including 7  
c) Any integer between 3.5 and 7, excluding 7  
d) The integer closest to the mean of 3.5 and 7  
View Answer

Answer: a  
Explanation: The function random.randint() does not accept a decimal value as a parameter. Hence the function shown above will throw an error.

8. Which of the following functions helps us to randomize the items of a list?  
a) seed  
b) randomise  
c) shuffle  
d) uniform  
View Answer

Answer: c  
Explanation: The function shuffle, which is included in the random module, helps us to randomize the items of a list. This function takes the list as a parameter.

9. What is the output of the code shown below?

random.seed(3)

random.randint(1,5)

2

random.seed(3)

random.randint(1,5)

a) 3  
b) 2  
c) Any integer between 1 and 5, including 1 and 5  
d) Any integer between 1 and 5, excluding 1 and 5  
View Answer

Answer: b  
Explanation: We use the seed function when we want to use the same random umber once again in our program. Hence the output of the code shown above will be 2, since 2 was generated previously following which we used the seed function.

10. What is the interval of the value generated by the function random.random(), assuming that the random module has already been imported?  
a) (0,1)  
b) (0,1]  
c) [0,1]  
d) [0,1)  
View Answer

Answer: d  
Explanation: The function random.random() generates a random value in the interval [0,1), that is, including zero but excluding one.

11. Which of the following is a possible outcome of the function shown below?

random.randrange(0,91,5)

a) 10  
b) 18  
c) 79  
d) 95  
View Answer

Answer: a  
Explanation: The function shown above will generate an output which is a multiple of 5 and is between 0 and 91. The only option which satisfies these criteria is 10. Hence the only possible output of this function is 10.

12. Both the functions randint and uniform accept \_\_\_\_\_\_\_\_\_\_\_\_ parameters.  
a) 0  
b) 1  
c) 3  
d) 2  
View Answer

Answer: c  
Explanation: Both of these functions, that is, randint and uniform are included in the random module and both of these functions accept 3 parameters. For example: random.uniform(self,a,b) where ‘a’ and ‘b’ specify the range and self is an imaginary parameter.

13. The randrange function returns only an integer value. Sate whether true or false.  
a) True  
b) False  
View Answer

Answer: a  
Explanation: The function randrange returns only an integer value. Hence this statement is true.

14. Which of the following options is the possible outcome of the function shown below?

random.randrange(1,100,10)

a) 32  
b) 67  
d) 91  
d) 80  
View Answer

Answer: c  
Explanaton: The output of this function can be any value which is a multiple of 10, plus 1. Hence a value like 11, 21, 31, 41…91 can be the output. Also, the value should necessarily be between 1 and 100. The only option which satisfies this criteria is 91.

15. What is the output of this function, assuming that the random library has already been included?

random.shuffle[1,2,24]

a) Randomized list containing the same numbers in any order  
b) The same list, that is [1,2,24].  
c) A list containing any random numbers between 1 and 24  
d) Error  
View Answer

Answer: d  
Explanation: The function shown above will result in an error because this is the incorrect syntax for the usage of the function shuffle(). The list should be previously declared and then passed to this function to get an output.  
An example of the correct syntax:  
>>> l=[‘a’,’b’,’c’,’d’].  
>>> random.shuffle(l)  
>>> print(l)

1. What the does random.seed(3) return?  
a) True  
b) None  
c) 3  
d) 1  
View Answer

Answer: b  
Explanation: The function random.seed() always returns a None.

2. Which of the following cannot be returned by random.randrange(4)?  
a) 0  
b) 3  
c) 2.3  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: Only integers can be returned.

3. Which of the following is equivalent to random.randrange(3)?  
a) range(3)  
b) random.choice(range(0, 3))  
c) random.shuffle(range(3))  
d) random.select(range(3))  
View Answer

Answer: b  
Explanation: It returns one number from the given range.

4. The function random.randint(4) can return only one of the following values. Which?  
a) 4  
b) 3.4  
c) error  
d) 5  
View Answer

Answer: c  
Explanation: Error, the function takes two arguments.

5. Which of the following is equivalent to random.randint(3, 6)?  
a) random.choice([3, 6])  
b) random.randrange(3, 6)  
c) 3 + random.randrange(3)  
d) 3 + random.randrange(4)  
View Answer

Answer: d  
Explanation: random.randint(3, 6) can return any one of 3, 4, 5 and 6.

6. Which of the following will not be returned by random.choice(“1 ,”)?  
a) 1  
b) (space)  
c) ,  
d) none of the mentioned  
View Answer

Answer: d  
Explanation: Any of the characters present in the string may be returned.

7. Which of the following will never be displayed on executing print(random.choice({0: 1, 2: 3}))?  
a) 0  
b) 1  
c) KeyError: 1  
d) none of the mentioned  
View Answer

Answer: a  
Explanation: It will not print 0 but dict[0] i.e. 1 may be printed.

8. What does random.shuffle(x) do when x = [1, 2, 3]?  
a) error  
b) do nothing, it is a placeholder for a function that is yet to be implemented  
c) shuffle the elements of the list in-place  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: The elements of the list passed to it are shuffled in-place.

9. Which type of elements are accepted by random.shuffle()?  
a) strings  
b) lists  
c) tuples  
d) integers  
View Answer

Answer: b  
Explanation: Strings and tuples are immutable and an integer has no len().

10. What is the range of values that random.random() can return?  
a) [0.0, 1.0].  
b) (0.0, 1.0].  
c) (0.0, 1.0)  
d) [0.0, 1.0)  
View Answer

Answer: d  
Explanation: Any number that is greater than or equal to 0.0 and lesser than 1.0 can be returned.