Python Questions and Answers – Global vs Local Variables – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Global vs Local Variables – 2”.

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.

Python Questions and Answers – Recursion

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Recursion”.

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

Answer: b
Explanation: In recursion, the function calls itself till the base condition is reached whereas iteration means repetition of process for example in for-loops.

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.

Python Questions and Answers – Shallow copy vs Deep copy

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Shallow copy vs Deep copy”.

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.

Python Questions and Answers – Functional Programming Tools

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Functional Programming Tools”.

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.

Python Questions and Answers – Regular Expressions

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Regular Expressions”.

1. Which module in Python supports regular expressions?

a) re

b) regex

c) pyregex

d) none of the mentioned

View Answer

Answer: a
Explanation: re is a part of the standard library and can be imported using: import re.

2. Which of the following creates a pattern object?

a) re.create(str)

b) re.regex(str)

c) re.compile(str)

d) re.assemble(str)

View Answer

Answer: c
Explanation: It converts a given string into a pattern object.

3. What does the function re.match do?

a) matches a pattern at the start of the string

b) matches a pattern at any position in the string

c) such a function does not exist

d) none of the mentioned

View Answer

Answer: a
Explanation: It will look for the pattern at the beginning and return None if it isn't found.

4. What does the function re.search do?

a) matches a pattern at the start of the string

b) matches a pattern at any position in the string

c) such a function does not exist

d) none of the mentioned

View Answer

Answer: b
Explanation: It will look for the pattern at any position in the string.

5. What is the output of the following?

```
sentence = 'we are humans'
matched = re.match(r'(.*) (.*?) (.*)', sentence)
print(matched.groups())
```

a) ('we', 'are', 'humans')

b) (we, are, humans)

c) ('we', 'humans')

d) 'we are humans'

View Answer

Answer: a
Explanation: This function returns all the subgroups that have been matched.

6. What is the output of the following?

```
sentence = 'we are humans'
matched = re.match(r'(.*) (.*?) (.*)', sentence)
print(matched.group())
```

a) ('we', 'are', 'humans')

b) (we, are, humans)

c) ('we', 'humans')

d) 'we are humans'

View Answer

Answer: d
Explanation: This function returns the entire match.

7. What is the output of the following?

```
sentence = 'we are humans'
matched = re.match(r'(.*) (.*?) (.*)', sentence)
print(matched.group(2))
```

a) ‘are’

b) ‘we’

c) ‘humans’

d) ‘we are humans’

View Answer

Answer: c
Explanation: This function returns the particular subgroup.

8. What is the output of the following?

```
sentence = 'horses are fast'
regex = re.compile('(?P<animal>\w+) (?P<verb>\w+) (?P<adjective>\w+)')
matched = re.search(regex, sentence)
print(matched.groupdict())
```

a) {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

b) (‘horses’, ‘are’, ‘fast’)

c) ‘horses are fast’

d) ‘are’

View Answer

Answer: a
Explanation: This function returns a dictionary that contains all the mathches.

9. What is the output of the following?

```
sentence = 'horses are fast'
regex = re.compile('(?P<animal>\w+) (?P<verb>\w+) (?P<adjective>\w+)')
matched = re.search(regex, sentence)
print(matched.groups())
```

a) {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

b) (‘horses’, ‘are’, ‘fast’)

c) ‘horses are fast’

d) ‘are’

View Answer

Answer: b
Explanation: This function returns all the subgroups that have been matched.

10. What is the output of the following?

```
sentence = 'horses are fast'
regex = re.compile('(?P<animal>\w+) (?P<verb>\w+) (?P<adjective>\w+)')
matched = re.search(regex, sentence)
print(matched.group(2))
```

a) {‘animal’: ‘horses’, ‘verb’: ‘are’, ‘adjective’: ‘fast’}

b) (‘horses’, ‘are’, ‘fast’)

c) ‘horses are fast’

d) ‘are’

View Answer

Answer: d
Explanation: This function returns the particular subgroup.

Python Questions and Answers – Regular Expressions – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Regular Expressions – 1”.

1. The character Dot (that is, ‘.’) in the default mode, matches any character other than _____

a) caret

b) ampersand

c) percentage symbol

d) newline

View Answer

Answer: d
Explanation: The character Dot (that is, ‘,’) in the default mode, matches any character other than newline. If DOTALL flag is used, then it matches any character other than newline.

2. The expression a{5} will match _____ characters with the previous regular expression.

a) 5 or less

b) exactly 5

c) 5 or more

d) exactly 4

View Answer

Answer: b
Explanation: The character {m} is used to match exactly m characters to the previous regular expression. Hence the expression a{5} will match exactly 5 characters and not less than that.

3. _____ matches the start of the string.

_____ matches the end of the string.

a) ‘^’, ‘$’

b) ‘$’, ‘^’

c) ‘$’, ‘?’

d) ‘?’, ‘^’

View Answer

Answer: a
Explanation: ‘^’ (carat) matches the start of the string.
‘$’ (dollar sign) matches the end of the string.

4. Which of the following will result in an error?

a) >>> p = re.compile(“d”)

>>> p.search(“door”)

b) >>> p = re.escape(‘hello’)

c) >>> p = re.subn()

d) >>> p = re.purge()

View Answer

Answer: c
Explanation: The function re.subn() will result in an error. This is because subn() requires 3 positional arguments while we have entered none.

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

```
re.split('\W+', 'Hello, hello, hello.')
```

a) [‘Hello’, ‘hello’, ‘hello.’]

b) [‘Hello, ‘hello’, ‘hello’]

c) [‘Hello’, ‘hello’, ‘hello’, ‘.’]

d) [‘Hello’, ‘hello’, ‘hello’, ”]

View Answer

Answer: d
Explanation: In the code shown above, the function split() splits the string based on the pattern given as an argument in the parenthesis. Note: split will never split a string on an empty pattern match. Hence the output of this code is: [‘Hello’, ‘hello’, ‘hello’, ”].

6. What is the output of the following function?

```
re.findall("hello world", "hello", 1)
```

a) [“hello”]

b) [ ]

c) hello

d) hello world

View Answer

Answer: b
Explanation: The function findall returns the word matched if and only if both the pattern and the string match completely, that is, they are exactly the same. Observe the example shown below: >>> re.findall(“hello”, “hello”, 1) The output is: [‘hello’] Hence the output of the code shown in this question is [].

7. Choose the function whose output can be: <_sre.SRE_Match object; span=(4, 8), match='aaaa'>.

a) >>> re.search(‘aaaa’, “alohaaaa”, 0)

b) >>> re.match(‘aaaa’, “alohaaaa”, 0)

c) >>> re.match(‘aaa’, “alohaaa”, 0)

d) >>> re.search(‘aaa’, “alohaaa”, 0)

View Answer

Answer: a
Explanation: The output shown above is that of a search function, whose pattern is ‘aaaa’ and the string is that of 8 characters. The only option which matches all these criteria is:
>>> re.search(‘aaaa’, “alohaaaa”, 0)

8. Which of the following functions clears the regular expression cache?

a) re.sub()

b) re.pos()

c) re.purge()

d) re.subn()

View Answer

Answer: c
Explanation: The function which clears the regular expression cache is re.purge(). Note that this function takes zero positional arguments.

9. What is the output of the code shown?

```
import re
re.ASCII
```

a) 8

b) 32

c) 64

d) 256

View Answer

Answer: d
Explanation: The expression re.ASCII returns the total number of ASCII characters that are present, that is 256. This can also be abbreviated as re.A, which results in the same output (that is, 256).

10. Which of the following functions results in case in-sensitive matching?

a) re.A

b) re.U

c) re.I

d) re.X

View Answer

Answer: c
Explanation: The function re.I (that is, re.IGNORECASE) results in case-insensitive matching. That is, expressions such as [A-Z] will match lowercase characters too.

Python Questions and Answers – Regular Expressions – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Regular Expressions – 2”.

1. What is the output of the code shown?

```
re.compile('hello', re.X)
```

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’]

b) re.compile(‘hello’, re.VERBOSE)

c) Error

d) Junk value

View Answer

Answer: b
Explanation: The compile function compiles a pattern of regular expression into an object of regular expression. Re.X is a flag which is also used as re.VERBOSE. Hence the output of this code is: re.compile(‘hello’, re.VERBOSE).

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

```
re.split('[a-c]', '0a3B6', re.I)
```

a) Error

b) [‘a’, ‘B’]

c) [‘0’, ‘3B6’]

d) [‘a’]

View Answer

Answer: c
Explanation: The function re.split() splits the string on the basis of the pattern given in the parenthesis. Since we have used the flag e.I (that is, re.IGNORECASE), the output is: [‘0’, ‘3B6’].

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

```
re.sub('morning', 'evening', 'good morning')
```

a) ‘good evening’

b) ‘good’

c) ‘morning’

d) ‘evening’

View Answer

Answer: a
Explanation: The code shown above first searches for the pattern ‘morning’ in the string ‘good morning’ and then replaces this pattern with ‘evening’. Hence the output of this code is: ‘good evening’.

4. The function re.error raises an exception if a particular string contains no match for the given pattern. State whether true or false.

a) True

b) False

View Answer

Answer: b
Explanation: The function re.error raises an exception when a string passed to one of its functions here is not a valid regular expression. It does not raise an exception if a particular string does not contain a match for the given pattern.

5. The output of the code shown below:

```
re.escape('new**world')
```

a) ‘new world’

b) ‘new**world’

c) ‘**’

d) ‘new’, ‘*’, ‘*’, ‘world’

View Answer

Answer: b
Explanation: The function re.escape escapes all the characters in the pattern other than ASCII letters and numbers. Hence the output of the code shown above is: ‘new**world’.

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

```
re.fullmatch('hello', 'hello world')
```

a) No output

b) []

c) <_sre.SRE_Match object; span=(0, 5), match='hello'>

d) Error

View Answer

Answer: a
Explanation: The function re.fullmatch applies the pattern to the entire string and returns an object if match is found and none if match in not found. In the code shown above, match is not found. Hence there is no output.

7. Choose the option wherein the two choices do not refer to the same option.

a) re.I

re.IGNORCASE

b) re.M

re.MULTILINE

c) re.X

re.VERBOSE

d) re.L

re.LOWERCASE

View Answer

Answer: d
Explanation: The function re.L is also written as re.LOCALE. There is no function such as re.LOWERCASE in the re module of Python.

8. The difference between the functions re.sub and re.subn is that re.sub returns a _____ whereas re.subn returns a _____

a) string, list

b) list, tuple

c) string, tuple

d) tuple, list

View Answer

Answer: c
Explanation: The difference the functions re.sub and re.subn is that re.sub returns a string whereas re.subn returns a tuple.

9. The output of the code shown below is:

```
re.split('mum', 'mumbai*', 1)
```

a) Error

b) [”, ‘bai*’]

c) [”, ‘bai’]

d) [‘bai*’]

View Answer

Answer: b
Explanation: The code shown above splits the string based on the pattern given as an argument. Hence the output of the code is: [”, ‘bai*’].

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

```
re.findall('good', 'good is good')
re.findall('good', 'bad is good')
```

a) [‘good’, ‘good’]

[‘good’]

b) (‘good’, ‘good’)

(good)

c) (‘good’)

(‘good’)

d) [‘good’]

[‘good’]

View Answer

Answer: a
Explanation: The function findall returns a list of all the non overlapping matches in a string. Hence the output of the first function is: [‘good’, ‘good’] and that of the second function is: [‘good’].

Python Questions and Answers – Regular Expressions – 4

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Regular Expressions – 4”.

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

```
re.split(r'(a)(t)', 'Maths is a difficult subject')
```

a) [‘M a t h s i s a d i f f i c u l t s u b j e c t’]

b) [‘Maths’, ‘is’, ‘a’, ‘difficult’, ‘subject’]

c) ‘Maths is a difficult subject’

d) [‘M’, ‘a’, ‘t’, ‘hs is a difficult subject’]

View Answer

Answer: d
Explanation: The code shown above demonstrates the use of the function re.match. The first argument of this function specifies the pattern. Since the pattern contains groups, those groups are incorporated in the resultant list as well. Hence the output of the code shown above is [‘M’, ‘a’, ‘t’, ‘hs is a difficult subject’].

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

```
CODE 1
>>> re.split(r'(a)(t)', 'The night sky')
CODE 2
>>> re.split(r'\s+', 'The night sky')
```

a) True

b) False

View Answer

Answer: b
Explanation: The output of the first code is: [‘The night sky’] whereas the output of the second code is:[‘The’, ‘night’, ‘sky’]. Clearly, the outputs of the two codes are different. Hence the statement given above is a false one.

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

```
import re
s = 'abc123 xyz666 lmn-11 def77'
re.sub(r'\b([a-z]+)(\d+)', r'\2\1:', s)
```

a) ‘123abc: 666xyz: lmn-11 77def:’

b) ‘77def: lmn-11: 666xyz: 123abc’

c) ‘abc123:’, ‘xyz666:’, ‘lmn-11:’, ‘def77:’

d) ‘abc123: xyz666: lmn-11: def77’

View Answer

Answer: a
Explanation: The function re.sub returns a string produced by replacing every non overlapping occurrence of the first argument with the second argument in the third argument. Hence the output is: ‘123abc: 666xyz: lmn-11 77def:’

4. What is the output of the code shown?

```
re.subn('A', 'X', 'AAAAAA', count=4)
```

a) ‘XXXXAA, 4’

b) (‘AAAAAA’, 4)

c) (‘XXXXAA’, 4)

d) ‘AAAAAA, 4’

View Answer

Answer: c
Explanation: The line of code shown above demonstrates the function re.subn. This function is very similar to the function re.sub except that in the former, a tuple is returned instead of a string. The output of the code shown above is: (‘XXXXAA’, 4).

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

```
n = re.sub(r'\w+', 'Hello', 'Cats and dogs')
```

a) Hello

Hello

Hello

b) ‘Hello Hello Hello’

c) [‘Hello’, ‘Hello’, ‘Hello’]

d) (‘Hello’, ‘Hello’, ‘Hello’)

View Answer

Answer: b
Explanation: The code shown above demonstrates the function re.sub. Since the string given as an argument consists of three words. The output of the code is: ‘Hello Hello Hello’. Had the string consisted of 4 words, the output would be: ‘Hello Hello Hello Hello’

6. What is the output of the following code?

```
w = re.compile('[A-Za-z]+')
w.findall('It will rain today')
```

a) ‘It will rain today’

b) (‘It will rain today’)

c) [‘It will rain today’]

d) [‘It’, ‘will’, ‘rain’, ‘today’]

View Answer

Answer: d
Explanation: The code shown above demonstrates the function re.findall. Since all the words in the string match the criteria, the output of the code is: [‘It’, ‘will’, ‘rain’, ‘today’].

7. In the functions re.search.start(group) and re.search.end(group), if the argument groups not specified, it defaults to _____

a) Zero

b) None

c) One

d) Error

View Answer

Answer: a
Explanation: In the functions re.search.start(group) and re.search.end(group), if the argument groups not specified, it defaults to Zero.

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

```
re.split(r'\s+', 'Chrome is better than explorer', maxspilt=3)
```

a) [‘Chrome’, ‘is’, ‘better’, ‘than’, ‘explorer’]

b) [‘Chrome’, ‘is’, ‘better’, ‘than explorer’]

c) (‘Chrome’, ‘is’, ‘better’, ‘than explorer’)

d) ‘Chrome is better’ ‘than explorer’

View Answer

Answer: b
Explanation: The code shown above demonstrates the use of the function re.split, including the use of maxsplit. Since maxsplit is equal to 3, the output of the code shown above is:[‘Chrome’, ‘is’, ‘better’, ‘than explorer’]

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

```
a=re.compile('[0-9]+')
a.findall('7 apples and 3 mangoes')
```

a) [‘apples’ ‘and’ ‘mangoes’]

b) (7, 4)

c) [‘7’, ‘4’]

d) Error

View Answer

Answer: c
Explanation: The code shown above demonstrates the use of the functions re.compile and re.findall. Since we have specified in the code that only digits from 0-9 be found, hence the output of this code is: [‘7’, ‘4’].

Python Questions and Answers – Regular Expressions – 5

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Regular Expressions – 5”.

1. Which of the following functions returns a dictionary mapping group names to group numbers?

a) re.compile.group

b) re.compile.groupindex

c) re.compile.index

d) re.compile.indexgroup

View Answer

Answer: b
Explanation: The function re.compile.groupindex returns a dictionary mapping group names to group numbers.

2. Which of the following statements regarding the output of the function re.match is incorrect?

a) ‘pq*’ will match ‘pq’

b) ‘pq?’ matches ‘p’

c) ‘p{4}, q’ does not match ‘pppq’

d) ‘pq+’ matches ‘p’

View Answer

Answer: d
Explanation: All of the above statements are correct except that ‘pq+’ match ‘p’. ‘pq+’ will match ‘p’ followed by any non-zero number of q’s, but it will not match ‘p’.

3. The snippet of code shown below results in an error. State whether true or false.

```
c=re.compile(r'(\d+)(\[A-Z]+)([a-z]+)')
c.groupindex
```

a) True

b) False

View Answer

Answer: b
Explanation: In the code shown above, none of the group names match the group numbers. In such a case, no error is thrown. The output of the code is an empty dictionary, that is, {}.

4. Which of the following functions does not accept any argument?

a) re.purge

b) re.compile

c) re.findall

d) re.match

View Answer

Answer: a
Explanation: The function re.purge is used to clear the cache and it does not accept any arguments.

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

```
a = re.compile('0-9')
a.findall('3 trees')
```

a) []

b) [‘3’]

c) Error

d) [‘trees’]

View Answer

Answer: c
Explanation: The output of the code shown above is an empty list. This is due to the way the arguments have been passed to the function re.compile. Carefully read the code shown below in order to understand the correct syntax:
>>> a = re.compile(‘[0-9]’)

>>> a.findall(‘3 trees’)
[‘3’].

6. Which of the following lines of code will not show a match?

a) >>> re.match(‘ab*’, ‘a’)

b) >>> re.match(‘ab*’, ‘ab’)

c) >>> re.match(‘ab*’, ‘abb’)

d) >>> re.match(‘ab*’, ‘ba’)

View Answer

Answer: d
Explanation: In the code shown above, ab* will match to ‘a’ or ‘ab’ or ‘a’ followed by any number of b’s. Hence the only line of code from the above options which does not result in a match is: >>> re.match(‘ab*’, ‘ba’).

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

```
m = re.search('a', 'The blue umbrella')
m.re.pattern
```

a) {}

b) ‘The blue umbrella’

c) ‘a’

d) No output

View Answer

Answer: c
Explanation: The PatternObject is used to produce the match. The real regular expression pattern string must be retrieved from the PatternObject’s pattern method. Hence the output of this code is: ‘a’.

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

```
re.sub('Y', 'X', 'AAAAAA', count=2)
```

a) ‘YXAAAA’

b) (‘YXAAAA’)

c) (‘AAAAAA’)

d) ‘AAAAAA’

View Answer

Answer: d
Explanation: The code shown above demonstrates the function re.sub, which returns a string. The pattern specified is substituted in the string and returned. Hence the output of the code shown above is: ‘AAAAAA’.

Python Questions and Answers – Files – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “files”.

1. To open a file c:\scores.txt for reading, we use

a) infile = open(“c:\scores.txt”, “r”)

b) infile = open(“c:\\scores.txt”, “r”)

c) infile = open(file = “c:\scores.txt”, “r”)

d) infile = open(file = “c:\\scores.txt”, “r”)

View Answer

Answer: b
Explanation: Execute help(open) to get more details.

2. To open a file c:\scores.txt for writing, we use

a) outfile = open(“c:\scores.txt”, “w”)

b) outfile = open(“c:\\scores.txt”, “w”)

c) outfile = open(file = “c:\scores.txt”, “w”)

d) outfile = open(file = “c:\\scores.txt”, “w”)

View Answer

Answer: b
Explanation: w is used to indicate that file is to be written to.

3. To open a file c:\scores.txt for appending data, we use

a) outfile = open(“c:\\scores.txt”, “a”)

b) outfile = open(“c:\\scores.txt”, “rw”)

c) outfile = open(file = “c:\scores.txt”, “w”)

d) outfile = open(file = “c:\\scores.txt”, “w”)

View Answer

Answer: a
Explanation: a is used to indicate that data is to be apended.

4. Which of the following statements are true?

a) When you open a file for reading, if the file does not exist, an error occurs

b) When you open a file for writing, if the file does not exist, a new file is created

c) When you open a file for writing, if the file exists, the existing file is overwritten with the new file

d) All of the mentioned

View Answer

Answer: d
Explanation: The program will throw an error.

5. To read two characters from a file object infile, we use

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

View Answer

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

6. To read the entire remaining contents of the file as a string from a file object infile, we use

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

View Answer

Answer: b
Explanation: read function is used to read all the lines in a file.

7. What is the output?

```
1. f = None
2. for i in range (5):
3.     with open("data.txt", "w") as f:
4.         if i > 2:
5.             break
6. print(f.closed)
```

a) True

b) False

c) None

d) Error

View Answer

Answer: a
Explanation: The WITH statement when used with open file guarantees that the file object is closed when the with block exits.

8. To read the next line of the file from a file object infile, we use

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

View Answer

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

9. To read the remaining lines of the file from a file object infile, we use

a) infile.read(2)

b) infile.read()

C) infile.readline()

d) infile.readlines()

View Answer

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

10. The readlines() method returns

a) str

b) a list of lines

c) a list of single characters

d) a list of integers

View Answer

Answer: b
Explanation: Every line is stored in a list and returned.

Python Questions and Answers – Files – 2

This set of Python Certification Questions & Answers focuses on “Files”.

1. Which are the two built-in functions to read a line of text from standard input, which by default comes from the keyboard?

a) Raw_input & Input

b) Input & Scan

c) Scan & Scanner

d) Scanner

View Answer

Answer: a
Explanation: Python provides two built-in functions to read a line of text from standard input, which by default comes from the keyboard. These functions are:
raw_input and input

2. What is the output of this program?

```
1. str = raw_input("Enter your input: ");
2. print "Received input is : ", str
```

a) Enter your input: Hello Python

Received input is : Hello Python

b) Enter your input: Hello Python

Received input is : Hello

c) Enter your input: Hello Python

Received input is : Python

d) None of the mentioned

View Answer

Answer: a
Explanation: The raw_input([prompt]) function reads one line from standard input and returns it as a string. This would prompt you to enter any string and it would display same string on the screen. When I typed “Hello Python!”

3. What is the output of this program?

```
1. str = input("Enter your input: ");
2. print "Received input is : ", str
```

a) Enter your input: [x*5 for x in range(2,10,2)].

Received input is : [x*5 for x in range(2,10,2)].

b) Enter your input: [x*5 for x in range(2,10,2)].

Received input is : [10, 30, 20, 40].

c) Enter your input: [x*5 for x in range(2,10,2)].

Received input is : [10, 10, 30, 40].

d) None of the mentioned

View Answer

Answer: a
Explanation: None.

4. Which one of the following is not attributes of file

a) closed

b) softspace

c) rename

d) mode

View Answer

Answer: c
Explanation: rename is not the attribute of file rest all are files attributes.
Attribute Description
file.closed Returns true if file is closed, false otherwise.
file.mode Returns access mode with which file was opened.
file.name Returns name of the file.
file.softspace Returns false if space explicitly required with print, true otherwise.

5. What is the use of tell() method in python?

a) tells you the current position within the file

b) tells you the end position within the file

c) tells you the file is opened or not

d) none of the mentioned

View Answer

Answer: a
Explanation: The tell() method tells you the current position within the file; in other words, the next read or write will occur at that many bytes from the beginning of the file.

6. What is the current syntax of rename() a file?

a) rename(current_file_name, new_file_name)

b) rename(new_file_name, current_file_name,)

c) rename(()(current_file_name, new_file_name))

d) none of the mentioned

View Answer

Answer: a
Explanation: This is the correct syntax which has shown below.
rename(current_file_name, new_file_name)

7. What is the current syntax of remove() a file?

a) remove(file_name)

b) remove(new_file_name, current_file_name,)

c) remove(() , file_name))

d) none of the mentioned

View Answer

Answer: a
Explanation: remove(file_name)

8. What is the output of this program?

```
1. fo = open("foo.txt", "rw+")
2. print "Name of the file: ", fo.name
3.
4. # Assuming file has following 5 lines
5. # This is 1st line
6. # This is 2nd line
7. # This is 3rd line
8. # This is 4th line
9. # This is 5th line
10.
11. for index in range(5):
12.    line = fo.next()
13.    print "Line No %d - %s" % (index, line)
14.
15. # Close opend file
16. fo.close()
```

a) Compilation Error

b) Syntax Error

c) Displays Output

d) None of the mentioned

View Answer

Answer: c
Explanation: It displays the output as shown below. The method next() is used when a file is used as an iterator, typically in a loop, the next() method is called repeatedly. This method returns the next input line, or raises StopIteration when EOF is hit.
Output:
Name of the file: foo.txt
Line No 0 – This is 1st line

Line No 1 – This is 2nd line

Line No 2 – This is 3rd line

Line No 3 – This is 4th line

Line No 4 – This is 5th line

9. What is the use of seek() method in files?

a) sets the file's current position at the offset

b) sets the file's previous position at the offset

c) sets the file's current position within the file

d) none of the mentioned

View Answer

Answer: a
Explanation: Sets the file's current position at the offset. The method seek() sets the file's current position at the offset.
Following is the syntax for seek() method:
fileObject.seek(offset[, whence])

Parameters

offset — This is the position of the read/write pointer within the file.

whence — This is optional and defaults to 0 which means absolute file positioning, other values are 1 which means seek relative to the current position and 2 means seek relative to the file's end.

10. What is the use of truncate() method in file?

a) truncates the file's size

b) deletes the content of the file

c) deletes the file's size

d) none of the mentioned

View Answer

Answer: a
Explanation: The method truncate() truncates the file's size. Following is the syntax for truncate() method:
fileObject.truncate([size])

Parameters

size — If this optional argument is present, the file is truncated to (at most) that size.

Python Questions and Answers – Files – 3

This set of Python Scripting Questions & Answers focuses on “Files”.

1. Which is/are the basic I/O connections in file?

a) Standard Input

b) Standard Output

c) Standard Errors

d) All of the mentioned

View Answer

Answer: d
Explanation: Standard input, standard output and standard error. Standard input is the data that goes to the program. The standard input comes from a keyboard. Standard output is where we print our data with the print keyword. Unless redirected, it is the terminal console. The standard error is a stream where programs write their error messages. It is usually the text terminal.

2. What is the output of this program?

```
1. import sys
2. print 'Enter your name: ',
3. name = ''
4. while True:
5.    c = sys.stdin.read(1)
6.    if c == '\n':
7.       break
8.    name = name + c
9.
10. print 'Your name is:', name
```

If entered name is

sanfoundry

a) sanfoundry

b) sanfoundry, sanfoundry

c) San

d) None of the mentioned

View Answer

Answer: a
Explanation: In order to work with standard I/O streams, we must import the sys module. The read() method reads one character from the standard input. In our example we get a promt saying “Enter your name”. We enter our name and press enter. The enter key generates the new line character: \n.
Output:
Enter your name: sanfoundry
Your name is: sanfoundry

3. What is the output of this program?

```
1. import sys
```

```
2. sys.stdout.write(' Hello\n')
3. sys.stdout.write('Python\n')
```

a) Compilation Error

b) Runtime Error

c) Hello Python

d) Hello

Python

View Answer

Answer: d
Explanation: None
Output:
Hello
Python

4. Which of the following mode will refer to binary data?

a) r

b) w

c) +

d) b

View Answer

Answer:d
Explanation: Mode Meaning is as explained below:
r Reading
w Writing
a Appending
b Binary data
+ Updating.

5. What is the pickling?

a) It is used for object serialization

b) It is used for object de-serialization

c) None of the mentioned

d) All of the mentioned

View Answer

Answer: a
Explanation: Pickle is the standard mechanism for object serialization. Pickle uses a simple stack-based virtual machine that records the instructions used to reconstruct the object. This makes pickle vulnerable to security risks by malformed or maliciously constructed data, that may cause the deserializer to import arbitrary modules and instantiate any object.

6. What is unpickling?

a) It is used for object serialization

b) It is used for object deserialization

c) None of the mentioned

d) All of the mentioned

View Answer

Answer: b
Explanation: We have been working with simple textual data. What if we are working with objects rather than simple text? For such situations, we can use the pickle module. This module serializes Python objects. The Python objects are converted into byte streams and written to text files. This process is called pickling. The inverse operation, reading from a file and reconstructing objects is called deserializing or unpickling.

7. What is the correct syntax of open() function?

a) file = open(file_name [, access_mode][, buffering])

b) file object = open(file_name [, access_mode][, buffering])

c) file object = open(file_name)

d) none of the mentioned

View Answer

Answer: b
Explanation: Open() function correct syntax with the parameter details as shown below:
file object = open(file_name [, access_mode][, buffering])
Here is parameters' detail:
file_name: The file_name argument is a string value that contains the name of the file that you want to access.
access_mode: The access_mode determines the mode in which the file has to be opened, i.e., read, write, append, etc. A complete list of possible values is given below in the table. This is optional parameter and the default file access mode is read (r).
buffering: If the buffering value is set to 0, no buffering will take place. If the buffering value is 1, line buffering will be performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action will be performed with the indicated buffer size. If negative, the buffer size is the system default(default behavior).

8. What is the output of this program?

```
1. fo = open("foo.txt", "wb")
2. print "Name of the file: ", fo.name
3. fo.flush()
4. fo.close()
```

a) Compilation Error

b) Runtime Error

c) No Output

d) Flushes the file when closing them

View Answer

Answer: d
Explanation: The method flush() flushes the internal buffer. Python automatically flushes the files when closing them. But you may want to flush the data before closing any file.

9. Correct syntax of file.writelines() is?

a) file.writelines(sequence)

b) fileObject.writelines()

c) fileObject.writelines(sequence)

d) none of the mentioned

View Answer

Answer: c
Explanation: The method writelines() writes a sequence of strings to the file. The sequence can be any iterable object producing strings, typically a list of strings. There is no return value.
Syntax
Following is the syntax for writelines() method:
fileObject.writelines(sequence).

10. Correct syntax of file.readlines() is?

a) fileObject.readlines(sizehint);

b) fileObject.readlines();

c) fileObject.readlines(sequence)

d) none of the mentioned

View Answer

Answer: a
Explanation: The method readlines() reads until EOF using readline() and returns a list containing the lines. If the optional sizehint argument is present, instead of reading up to EOF, whole lines totalling approximately sizehint bytes (possibly after rounding up to an internal buffer size) are read.
Syntax
Following is the syntax for readlines() method:
fileObject.readlines(sizehint);
Parameters
sizehint — This is the number of bytes to be read from the file.

Python Questions and Answers – Files – 4

This set of Python Scripting Interview Questions & Answers focuses on “Files”.

1. In file handling, what does this terms means “r, a”?

a) read, append

b) append, read

c) all of the mentioned

d) none of the the mentioned

View Answer

Answer: a
Explanation: r- reading, a-appending.

2. What is the use of “w” in file handling?

a) Read

b) Write

c) Append

d) None of the the mentioned

View Answer

Answer: b
Explanation: This opens the file for writing. It will create the file if it doesn’t exist, and if it does, it will overwrite it.
fh = open(“filename_here”, “w”).

3. What is the use of “a” in file handling?

a) Read

b) Write

c) Append

d) None of the the mentioned

View Answer

Answer: c
Explanation: This opens the fhe file in appending mode. That means, it will be open for writing and everything will be written to the end of the file.
fh =open(“filename_here”, “a”).

4. Which function is used to read all the characters?

a) Read()

b) Readcharacters()

c) Readall()

d) Readchar()

View Answer

Answer: a
Explanation: The read function reads all characters fh = open(“filename”, “r”)
content = fh.read().

5. Which function is used to read single line from file?

a) Readline()

b) Readlines()

c) Readstatement()

d) Readfullline()

View Answer

Answer: b
Explanation: The readline function reads a single line from the file fh = open(“filename”, “r”)
content = fh.readline().

6. Which function is used to write all the characters?

a) write()

b) writecharacters()

c) writeall()

d) writechar()

View Answer

Answer: a
Explanation: To write a fixed sequence of characters to a file
fh = open(“hello.txt”,”w”)
write(“Hello World”).

7. Which function is used to write a list of string in a file

a) writeline()

b) writelines()

c) writestatement()

d) writefullline()

View Answer

Answer: a
Explanation: With the writeline function you can write a list of strings to a file
fh = open(“hello.txt”, “w”)
lines_of_text = [“a line of text”, “another line of text”, “a third line”] fh.writelines(lines_of_text).

8. Which function is used to close a file in python?

a) Close()

b) Stop()

c) End()

d) Closefile()

View Answer

Answer: a
Explanation: f.close()to close it and free up any system resources taken up by the open file.

9. Is it possible to create a text file in python?

a) Yes

b) No

c) Machine dependent

d) All of the mentioned

View Answer

Answer: a
Explanation: Yes we can create a file in python. Creation of file is as shown below.
file = open(“newfile.txt”, “w”)
file.write(“hello world in the new file\n”)
file.write(“and another line\n”)
file.close().

10. Which of the following is modes of both writing and reading in binary format in file.?

a) wb+

b) w

c) wb

d) w+

View Answer

Answer: a
Explanation: Here is the description below
“w” Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
“wb” Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
“w+” Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
“wb+” Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.

Python Questions and Answers – Files – 5

This set of Python written test Questions & Answers focuses on “Files”.

1. Which of the following is not a valid mode to open a file?

a) ab

b) rw

c) r+

d) w+

View Answer

Answer: b
Explanation: Use r+, w+ or a+ to perform both read and write operations using a single file object.

2. What is the difference between r+ and w+ modes?

a) no difference

b) in r+ the pointer is initially placed at the beginning of the file and the pointer is at the end for w+

c) in w+ the pointer is initially placed at the beginning of the file and the pointer is at the end for r+

d) depends on the operating system

View Answer

Answer: b
Explanation: none.

3. How do you get the name of a file from a file object (fp)?

a) fp.name

b) fp.file(name)

c) self.__name__(fp)

d) fp.__name__()

View Answer

Answer: a
Explanation: name is an attribute of the file object.

4. Which of the following is not a valid attribute of a file object (fp)?

a) fp.name

b) fp.closed

c) fp.mode

d) fp.size

View Answer

Answer: d
Explanation: fp.size has not been implemented.

5. How do you close a file object (fp)?

a) close(fp)

b) fclose(fp)

c) fp.close()

d) fp.__close__()

View Answer

Answer: c
Explanation: close() is a method of the file object.

6. How do you get the current position within the file?

a) fp.seek()

b) fp.tell()

c) fp.loc

d) fp.pos

View Answer

Answer: b
Explanation: It gives the current position as an offset from the start of file.

7. How do you rename a file?

a) fp.name = ‘new_name.txt’

b) os.rename(existing_name, new_name)

c) os.rename(fp, new_name)

d) os.set_name(existing_name, new_name)

View Answer

Answer: b
Explanation: os.rename() is used to rename files.

8. How do you delete a file?

a) del(fp)

b) fp.delete()

c) os.remove('file')

d) os.delete('file')

View Answer

Answer: c
Explanation: os.remove() is used to delete files.

9. How do you change the file position to an offset value from the start?

a) fp.seek(offset, 0)

b) fp.seek(offset, 1)

c) fp.seek(offset, 2)

d) none of the mentioned

View Answer

Answer: a
Explanation: 0 indicates that the offset is with respect to the start.

10. What happens if no arguments are passed to the seek function?

a) file position is set to the start of file

b) file position is set to the end of file

c) file position remains unchanged

d) error

View Answer

Answer: d
Explanation: seek() takes at least one argument.

Python Questions and Answers – Operator Overloading

This set of Python Interview Questions and Answers for Experienced people focuses on "Operator Overloading"

1. Which function is called when the following code is executed?

```
f = foo()
format(f)
```

a) format()

b) __format__()

c) str()

d) __str__()

View Answer

Answer: d
Explanation: Both str(f) and format(f) call f.__str__().

2. Which of the following will print True?

```
a = foo(2)
b = foo(3)
print(a < b)
```

a)

```
class foo:
    def __init__(self, x):
        self.x = x
    def __lt__(self, other):
        if self.x < other.x:
            return False
        else:
            return True
```

b)

```
class foo:
    def __init__(self, x):
        self.x = x
    def __less__(self, other):
        if self.x > other.x:
            return False
        else:
            return True
```

c)

```
class foo:
    def __init__(self, x):
        self.x = x
    def __lt__(self, other):
        if self.x < other.x:
            return True
        else:
            return False
```

d)

```
class foo:
    def __init__(self, x):
        self.x = x
    def __less__(self, other):
        if self.x < other.x:
            return False
        else:
            return True
```

View Answer

Answer: c
Explanation: __lt__ overloads the < operator>.

3. Which function overloads the + operator?

a) __add__()

b) __plus__()

c) __sum__()

d) none of the mentioned

View Answer

Answer: a
Explanation: Refer documentation.

4. Which operator is overloaded by __invert__()?

a) !

b) ~

c) ^

d) –

View Answer

5. Which function overloads the == operator?

a) __eq__()

b) __equ__()

c) __isequal__()

d) none of the mentioned

View Answer

Answer: a
Explanation: The other two do not exist.

6. Which operator is overloaded by __lg__()?

a) <

b) >

c) !=

d) none of the mentioned

View Answer

Answer: d
Explanation: __lg__() is invalid.

7. Which function overloads the >> operator?

a) __more__()

b) __gt__()

c) __ge__()

d) none of the mentioned

View Answer

Answer: d
Explanation: __rshift__() overloads the >> operator.

8. Let A and B be objects of class Foo. Which functions are called when print(A + B) is executed?

a) __add__(), __str__()

b) __str__(), __add__()

c) __sum__(), __str__()

d) __str__(), __sum__()

View Answer

Answer: a
Explanation: The function __add__() is called first since it is within the bracket. The function __str__() is then called on the object that we recieved after adding A and B.

9. Which operator is overloaded by the __or__() function?

a) ||

b) |

c) //

d) /

View Answer

Answer: b
Explanation: The function __or__() overloads the bitwise OR operator |.

10. Which function overloads the // operator?

a) __div__()

b) __ceildiv__()

c) __floordiv__()

d) __truediv__()

View Answer

Answer: c
Explanation: __floordiv__() is for //.

Python Questions and Answers – Classes and Objects – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Classes and Objects – 1”.

1. _____ represents an entity in the real world with its identity and behaviour.

a) A method

b) An object

c) A class

d) An operator

View Answer

Answer: b
Explanation: An object represents an entity in the real world that can be distinctly identified. A class may define an object.

2. _____ is used to create an object.

a) class

b) constructor

c) User-defined functions

d) In-built functions

View Answer

Answer: b
Explanation: The values assigned by the constructor to the class members is used to create the object.

3. What is the output of the following code?

```
class test:
     def __init__(self,a="Hello World"):
         self.a=a

     def display(self):
         print(self.a)
obj=test()
obj.display()
```

a) The program has an error because constructor can’t have default arguments

b) Nothing is displayed

c) “Hello World” is displayed

d) The program has an error display function doesn’t have parameters

View Answer

Answer: c
Explanation: The program has no error. “Hello World” is displayed. Execute in python shell to verify.

4. What is setattr() used for?

a) To access the attribute of the object

b) To set an attribute

c) To check if an attribute exists or not

d) To delete an attribute

View Answer

Answer: b
Explanation: setattr(obj,name,value) is used to set an attribute. If attribute doesn’t exist, then it would be created.

5. What is getattr() used for?

a) To access the attribute of the object

b) To delete an attribute

c) To check if an attribute exists or not

d) To set an attribute

View Answer

Answer: a
Explanation: getattr(obj,name) is used to get the attribute of an object.

6. What is the output of the following code?

```
class change:
    def __init__(self, x, y, z):
        self.a = x + y + z

x = change(1,2,3)
y = getattr(x, 'a')
setattr(x, 'a', y+1)
print(x.a)
```

a) 6

b) 7

c) Error

d) 0

View Answer

Answer: b
Explanation: First, a=1+2+3=6. Then, after setattr() is invoked, x.a=6+1=7.

7. What is the output of the following code?

```
 class test:
     def __init__(self,a):
         self.a=a

     def display(self):
         print(self.a)
obj=test()
obj.display()
```

a) Runs normally, doesn’t display anything

b) Displays 0, which is the automatic default value

c) Error as one argument is required while creating the object

d) Error as display function requires additional argument

View Answer

Answer: c
Explanation: Since, the __init__ special method has another argument a other than self, during object creation, one argument is required. For example: obj=test(“Hello”)

8. Is the following piece of code correct?

```
>>> class A:
        def __init__(self,b):
```

```
                self.b=b
        def display(self):
                print(self.b)
>>> obj=A("Hello")
>>> del obj
```

a) True

b) False

View Answer

Answer: a
Explanation: It is possible to delete an object of the class. On further typing obj in the python shell, it throws an error because the defined object has now been deleted.

9. What is the output of the following code?

```
class test:
    def __init__(self):
        self.variable = 'Old'
        self.Change(self.variable)
    def Change(self, var):
        var = 'New'
obj=test()
print(obj.variable)
```

a) Error because function change can't be called in the __init__ function

b) 'New' is printed

c) 'Old' is printed

d) Nothing is printed

View Answer

Answer: c
Explanation: This is because strings are immutable. Hence any change made isn't reflected in the original string.

10. What is Instantiation in terms of OOP terminology?

a) Deleting an instance of class

b) Modifying an instance of class

c) Copying an instance of class

d) Creating an instance of class

View Answer

Answer: d
Explanation: Instantiation refers to creating an object/instance for a class.

11. What is the output of the following code?

```
class fruits:
    def __init__(self, price):
        self.price = price
obj=fruits(50)

obj.quantity=10
```

```
obj.bags=2

print(obj.quantity+len(obj.__dict__))
```

a) 12

b) 52

c) 13

d) 60

View Answer

Answer: c
Explanation: In the above code, obj.quantity has been initialised to 10. There are a total of three items in the dictionary, price, quantity and bags. Hence, len(obj.__dict__) is 3.

12. What is the output of the following code?

```
 class Demo:
    def __init__(self):
        pass

    def test(self):
        print(__name__)

obj = Demo()
obj.test()
```

a) Exception is thrown

b) __main__

c) Demo

d) test

View Answer

Answer: b
Explanation: Since the above code is being run not as a result of an import from another module, the variable will have value “__main__”.

Python Questions and Answers – Classes and Objects – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Classes and Objects – 2”.

1. The assignment of more than one function to a particular operator is _____

a) Operator over-assignment

b) Operator overriding

c) Operator overloading

d) Operator instance

View Answer

Answer: c
Explanation: The assignment of more than one function to a particular operator is called as operator overloading.

2. Which of the following is not a class method?

a) Non-static

b) Static

c) Bounded

d) Unbounded

View Answer

Answer: a
Explanation: The three different class methods in Python are static, bounded and unbounded methods.

3. What is the output of the following code?

```
    c.test=c.test+1
    k=k+1
class A:
    def __init__(self):
        self.test = 0
def main():
    Count=A()
    k=0

    for i in range(0,25):
        add(Count,k)
    print("Count.test=", Count.test)
    print("k =", k)
main()
```

a) Exception is thrown

b) Count.test=25

k=25

c) Count.test=25

k=0

d) Count.test=0

k=0

View Answer

Answer: c
Explanation: The program has no error. Here, test is a member of the class while k isn't. Hence test keeps getting incremented time while k remains 0.

4. Which piece of code creates an empty class?

a)

```
class A:
    return
```

b)

```
class A:
    pass
```

c)

```
class A:
```

d) It is not possible to create an empty class.

View Answer

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

5. Is the following piece of code valid?

```
class B(object):
  def first(self):
    print("First method called")
  def second():
    print("Second method called")
ob = B()
B.first(ob)
```

a) It isn't as the object declaration isn't right

b) It isn't as there isn't any __init__ method for initializing class members

c) Yes, this method of calling is called unbounded method call

d) Yes, this method of calling is called bounded method call

View Answer

Answer: c
Explanation: The method may be created in the method demonstrated in the code as well and this is called as the unbounded method call. Calling the method using obj.one() is the bounded method call.

6. What are the methods which begin and end with two underscore characters called?

a) Special methods

b) In-built methods

c) User-defined methods

d) Additional methods

View Answer

Answer: a
Explanation: Special methods like __init__ begin and end with two underscore characters.

7. Special methods need to be explicitly called during object creation. True or False?

a) True

b) False

View Answer

Answer: b
Explanation: Special methods are automatically called during object creation.

8. What is the output of the following code?

```
>>> class demo():
        def __repr__(self):
                return '__repr__ built-in function called'
        def __str__(self):
                return '__str__ built-in function called'
>>> s=demo()
>>> s
```

a) Error

b) Nothing is printed

c) __str__ called

d) __repr__ called

View Answer

Answer: d
Explanation: __repr__ is used for producing a string representation of an object's value that Python can evaluate. Execute in python shell to verify.

9. What is the output of the following code?

```
>>> class demo():
        def __repr__(self):
                return '__repr__ built-in function called'
        def __str__(self):
                return '__str__  built-in function called'
>>> s=demo()
>>> print(s)
```

a) __str__ called

b) __repr__ called

c) Error

d) Nothing is printed

View Answer

Answer: a
Explanation: __str__ is used for producing a string representation of an object's value that is most readable for humans. Execute in python shell to verify.

10. What is hasattr(obj,name) used for?

a) To access the attribute of the object

b) To delete an attribute

c) To check if an attribute exists or not

d) To set an attribute

View Answer

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

```
class stud:
   def __init__(self, roll_no, grade):
      self.roll_no = roll_no
```

```
      self.grade = grade
   def display (self):
      print("Roll no : ", self.roll_no,  ", Grade: ", self.grade)
stud1 = stud(34, 'S')
stud1.age=7
print(hasattr(stud1, 'age'))
```

a) Error as age isn't defined

b) True

c) False

d) 7

View Answer

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

12. What is delattr(obj,name) used for?

a) To print deleted attribute

b) To delete an attribute

c) To check if an attribute is deleted or not

d) To set an attribute

View Answer

Answer: b
Explanation: delattr(obj,name) deletes an attribute in a class.

13. __del__ method is used to destroy instances of a class. True or False?

a) True

b) False

View Answer

Answer: a
Explanation: ___del__ method acts as a destructor and is used to destroy objects of classes.

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

```
class stud:
   'Base class for all students'
   def __init__(self, roll_no, grade):
      self.roll_no = roll_no
      self.grade = grade
   def display (self):
      print("Roll no : ", self.roll_no,  ", Grade: ", self.grade)
print(student.__doc__)
```

a) Exception is thrown

b) __main__

c) Nothing is displayed

d) Base class for all students

View Answer

Answer: d
Explanation: ___doc__ built-in class attribute is used to print the class documentation string or none, if undefined.

15. What does print(Test.__name__) display (assuming Test is the name of the class) ?
a) ()
b) Exception is thrown
c) Test
d) __main__
View Answer

Answer: c
Explanation: __name__ built-in class attribute is used to display the class name.

Python Questions and Answers – Inheritance – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on "Inheritance – 1".

1. Which of the following best describes inheritance?
a) Ability of a class to derive members of another class as a part of its own definition
b) Means of bundling instance variables and methods in order to restrict access to certain class members
c) Focuses on variables and passing of variables to functions
d) Allows for implementation of elegant software that is well designed and easily modified
View Answer

Answer: a
Explanation: If the class definition is class B(A): then class B inherits the methods of class A. This is called inheritance.

2. Which of the following statements is wrong about inheritance?
a) Protected members of a class can be inherited
b) The inheriting class is called a subclass
c) Private members of a class can be inherited and accessed
d) Inheritance is one of the features of OOP
View Answer

Answer: c
Explanation: Any changes made to the private members of the class in the subclass aren't reflected in the original members.

3. What is the output of the following code?

```
class Demo:
```

```
    def __new__(self):
        self.__init__(self)
        print("Demo's __new__() invoked")
    def __init__(self):
        print("Demo's __init__() invoked")
class Derived_Demo(Demo):
    def __new__(self):
        print("Derived_Demo's __new__() invoked")
    def __init__(self):
        print("Derived_Demo's __init__() invoked")
def main():
    obj1 = Derived_Demo()
    obj2 = Demo()
main()
```

a)

Derived_Demo's __init__() invoked

Derived_Demo's __new__() invoked

Demo's __init__() invoked

Demo's __new__() invoked

b)

Derived_Demo's __new__() invoked

Demo's __init__() invoked

Demo's __new__() invoked

c)

Derived_Demo's __new__() invoked

Demo's __new__() invoked

d)

Derived_Demo's __init__() invoked

Demo's __init__() invoked

View Answer

Answer: b

Explanation: Since the object for the derived class is declared first, __new__() method of the derived class is invoked first, followed by the constructor and the __new__() method of main class.

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

```
class Test:
    def __init__(self):
        self.x = 0
class Derived_Test(Test):
    def __init__(self):
        self.y = 1
def main():
    b = Derived_Test()
    print(b.x,b.y)
main()
```

a) 0 1

b) 0 0

c) Error because class B inherits A but variable x isn't inherited

d) Error because when object is created, argument must be passed like Derived_Test(1)

View Answer

Answer: c
Explanation: Since the invoking method, Test.__init__(self), isn't present in the derived class, variable x can't be inherited.

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

```
class A():
    def disp(self):
        print("A disp()")
class B(A):
    pass
obj = B()
obj.disp()
```

a) Invalid syntax for inheritance

b) Error because when object is created, argument must be passed

c) Nothing is printed

d) A disp()

View Answer

Answer: d
Explanation: Class B inherits class A hence the function disp () becomes part of class B's definition. Hence disp() method is properly executed and the line is printed.

6. All subclasses are a subtype in object-oriented programming. Is the statement true or false?

a) True

b) False

View Answer

Answer: b
Explanation: A subtype is something that be substituted for and behave as its parent type. All subclass may not be a subtype in object-oriented programming.

7. When defining a subclass in Python that is meant to serve as a subtype, the subtype Python keyword is used. Is the statement true or false?

a) True

b) False

View Answer

Answer: b
Explanation: B is a subtype of B if instances of type B can substitute for instances of type A without affecting semantics.

8. Suppose B is a subclass of A, to invoke the __init__ method in A from B, what is the line of code you should write?

a) A.__init__(self)

b) B.__init__(self)

c) A.__init__(B)

d) B.__init__(A)

View Answer

Answer: a
Explanation: To invoke the __init__ method in A from B, either of the following should be written: A.__init__(self) or super().__init__(self).

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

```
class Test:
    def __init__(self):
        self.x = 0
class Derived_Test(Test):
    def __init__(self):
        Test.__init__(self)
        self.y = 1
def main():
    b = Derived_Test()
    print(b.x,b.y)
main()
```

a) Error because class B inherits A but variable x isn't inherited

b) 0 0

c) 0 1

d) Error, the syntax of the invoking method is wrong

View Answer

Answer: c
Explanation: Since the invoking method has been properly invoked, variable x from the main class has been properly inherited and it can also be accessed.

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

```
class A:
    def __init__(self, x= 1):
        self.x = x
class der(A):
    def __init__(self,y = 2):
        super().__init__()
        self.y = y
def main():
    obj = der()
    print(obj.x, obj.y)
main()
```

a) Error, the syntax of the invoking method is wrong

b) The program runs fine but nothing is printed

c) 1 0

d) 1 2

View Answer

Answer: d
Explanation: In the above piece of code, the invoking method has been properly implemented and hence x=1 and y=2.

11. What does built-in function type do in context of classes?

a) Determines the object name of any value

b) Determines the class name of any value

c) Determines class description of any value

d) Determines the file name of any value

View Answer

Answer: b
Explanation: For example: >>> type((1,)) gives .

12. Which of the following is not a type of inheritance?

a) Double-level

b) Multi-level

c) Single-level

d) Multiple

View Answer

Answer: a
Explanation: Multiple, multi-level, single-level and hierarchical inheritance are all types of inheritance.

13. What does built-in function help do in context of classes?

a) Determines the object name of any value

b) Determines the class identifiers of any value

c) Determines class description of any built-in type

d) Determines class description of any user-defined built-in type

View Answer

Answer: c
Explanation: help() usually gives information of the class on any built-in type or function.

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

```
class A:
    def one(self):
        return self.two()

    def two(self):
        return 'A'

class B(A):
```

```
    def two(self):
        return 'B'
obj1=A()
obj2=B()
print(obj1.two(),obj2.two())
```

a) A A

b) A B

c) B B

d) An exception is thrown

View Answer

Answer: b
Explanation: obj1.two() invokes the method two() in class A which returns ‘A’ and obj2.two() invokes the method two() in class B which returns ‘B’.

15. What type of inheritance is illustrated in the following piece of code?

```
class A():
    pass
class B():
    pass
class C(A,B):
    pass
```

a) Multi-level inheritance

b) Multiple inheritance

c) Hierarchical inheritance

d) Single-level inheritance

View Answer

Answer: b
Explanation: In multiple inheritance, two or more subclasses are derived from the superclass as shown in the above piece of code.

Python Questions and Answers – Inheritance – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Inheritance – 2”.

1. What type of inheritance is illustrated in the following piece of code?

```
class A():
    pass
class B(A):
    pass
class C(B):
    pass
```

a) Multi-level inheritance

b) Multiple inheritance

c) Hierarchical inheritance

d) Single-level inheritance

View Answer

Answer: a
Explanation: In multi-level inheritance, a subclass derives from another class which itself is derived from another class.

2. What does single-level inheritance mean?

a) A subclass derives from a class which in turn derives from another class

b) A single superclass inherits from multiple subclasses

c) A single subclass derives from a single superclass

d) Multiple base classes inherit a single derived class

View Answer

Answer: c
Explanation: In single-level inheritance, there is a single subclass which inherits from a single superclass. So the class definition of the subclass will be: class B(A): where A is the superclass.

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

```
class A:
     def __init__(self):
         self.__i = 1
         self.j = 5

     def display(self):
         print(self.__i, self.j)
class B(A):
     def __init__(self):
         super().__init__()
         self.__i = 2
         self.j = 7
c = B()
c.display()
```

a) 2 7

b) 1 5

c) 1 7

d) 2 5

View Answer

Answer: c
Explanation: Any change made in variable i isn't reflected as it is the private member of the superclass.

4. Which of the following statements isn't true?

a) A non-private method in a superclass can be overridden

b) A derived class is a subset of superclass

c) The value of a private variable in the superclass can be changed in the subclass

d) When invoking the constructor from a subclass, the constructor of superclass is

automatically invoked

View Answer

Answer: c
Explanation: If the value of a private variable in a superclass is changed in the subclass, the change isn't reflected.

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

```
class A:
    def __init__(self,x):
        self.x = x
    def count(self,x):
        self.x = self.x+1
class B(A):
    def __init__(self, y=0):
        A.__init__(self, 3)
        self.y = y
    def count(self):
        self.y += 1
def main():
    obj = B()
    obj.count()
    print(obj.x, obj.y)
main()
```

a) 3 0

b) 3 1

c) 0 1

d) An exception in thrown

View Answer

Answer: b
Explanation: Initially x=3 and y=0. When obj.count() is called, y=1.

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

```
>>> class A:
        pass
>>> class B(A):
        pass
>>> obj=B()
>>> isinstance(obj,A)
```

a) True

b) False

c) Wrong syntax for isinstance() method

d) Invalid method for classes

View Answer

Answer: a
Explanation: isinstance(obj,class) returns True if obj is an object class.

7. Which of the following statements is true?

a) The __new__() method automatically invokes the __init__ method

b) The __init__ method is defined in the object class

c) The __eq(other) method is defined in the object class

d) The __repr__() method is defined in the object class

View Answer

Answer: c
Explanation: The __eq(other) method is called if any comparison takes place and it is defined in the object class.

8. Method issubclass() checks if a class is a subclass of another class. True or False?

a) True

b) False

View Answer

Answer: a
Explanation: Method issubclass() returns True if a class is a subclass of another class and False otherwise.

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

```
class A:
    def __init__(self):
        self.__x = 1
class B(A):
    def display(self):
        print(self.__x)
def main():
    obj = B()
    obj.display()
main()
```

a) 1

b) 0

c) Error, invalid syntax for object declaration

d) Error, private class member can’t be accessed in a subclass

View Answer

Answer: d
Explanation: Private class members in the superclass can’t be accessed in the subclass.

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

```
class A:
    def __init__(self):
        self._x = 5
class B(A):
    def display(self):
        print(self._x)
def main():
    obj = B()
```

```
    obj.display()
main()
```

a) Error, invalid syntax for object declaration

b) Nothing is printed

c) 5

d) Error, private class member can’t be accessed in a subclass

View Answer

Answer: c
Explanation: The class member x is protected, not private and hence can be accessed by subclasses.

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

```
class A:
    def __init__(self,x=3):
        self._x = x
class B(A):
    def __init__(self):
        super().__init__(5)
    def display(self):
        print(self._x)
def main():
    obj = B()
    obj.display()

main()
```

a) 5

b) Error, class member x has two values

c) 3

d) Error, protected class member can’t be accessed in a subclass

View Answer

Answer: a
Explanation: The super() method re-assigns the variable x with value 5. Hence 5 is printed.

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

```
class A:
    def test1(self):
        print(" test of A called ")
class B(A):
    def test(self):
        print(" test of B called ")
class C(A):
    def test(self):
        print(" test of C called ")
class D(B,C):
    def test2(self):
        print(" test of D called ")
obj=D()
obj.test()
```

a) test of B called

test of C called

b) test of C called

test of B called

c) test of B called

d) Error, both the classes from which D derives has same method test()

View Answer

Answer: c
Explanation: Execute in Python shell to verify. If class D(B,C): is switched is class D(C,B): test of C is called.

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

```
class A:
    def test(self):
        print("test of A called")
class B(A):
    def test(self):
        print("test of B called")
        super().test()
class C(A):
    def test(self):
        print("test of C called")
        super().test()
class D(B,C):
    def test2(self):
        print("test of D called")
obj=D()
obj.test()
```

a) test of B called

test of C called

test of A called

b) test of C called

test of B called

c) test of B called

test of C called

d) Error, all the three classes from which D derives has same method test()

View Answer

Answer: a
Explanation: Since the invoking method, super().test() is called in the subclasses, all the three methods of test() in three different classes is called.

Python Questions and Answers – Polymorphism

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Polymorphism”.

1. Which of the following best describes polymorphism?

a) Ability of a class to derive members of another class as a part of its own definition

b) Means of bundling instance variables and methods in order to restrict access to certain class members

c) Focuses on variables and passing of variables to functions

d) Allows for objects of different types and behaviour to be treated as the same general type

View Answer

Answer: d

Explanation: Polymorphism is a feature of object-oriented programming languages. It allows for the implementation of elegant software that is well designed and easily modified.

2. What is the biggest reason for the use of polymorphism?

a) It allows the programmer to think at a more abstract level

b) There is less program code to write

c) The program will have a more elegant design, and will be easier to maintain and update

d) Program code takes up lesser space

View Answer

Answer: c

Explanation: Polymorphism allows for the implementation of elegant software.

3. What is the use of duck typing?

a) More restriction on the type values that can be passed to a given method

b) No restriction on the type values that can be passed to a given method

c) Less restriction on the type values that can be passed to a given method

d) Makes the program code smaller

View Answer

Answer: c

Explanation: In Python, any set of classes with a common set of methods can be treated similarly. This is called duck typing. Hence duck typing imposes lesser restrictions.

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

```
class A:
    def __str__(self):
        return '1'
class B(A):
    def __init__(self):
        super().__init__()
class C(B):
    def __init__(self):
        super().__init__()
def main():
    obj1 = B()
```

```
    obj2 = A()
    obj3 = C()
    print(obj1, obj2,obj3)
main()
```

a) 1 1 1

b) 1 2 3

c) '1' '1' '1'

d) An exception is thrown

View Answer

Answer: a
Explanation: The super().__init__() in the subclasses has been properly invoked and none of other subclasses return any other value. Hence 1 is returned each time the object is created and printed.

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

```
class Demo:
    def __init__(self):
        self.x = 1
    def change(self):
        self.x = 10
class Demo_derived(Demo):
    def change(self):
        self.x=self.x+1
        return self.x
def main():
    obj = Demo_derived()
    print(obj.change())

main()
```

a) 11

b) 2

c) 1

d) An exception is thrown

View Answer

Answer: d
Explanation: The derived class method change() overrides the base class method.

6. A class in which one or more methods are only implemented to raise an exception is called an abstract class. True or False?

a) True

b) False

View Answer

7. Overriding means changing behaviour of methods of derived class methods in the base class. Is the statement true or false?

a) True

b) False

View Answer

Answer: b
Explanation: Overriding means if there are two same methods present in the superclass and the subclass, the contents of the subclass method are executed.

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

```
class A:
    def __repr__(self):
        return "1"
class B(A):
    def __repr__(self):
        return "2"
class C(B):
    def __repr__(self):
        return "3"
o1 = A()
o2 = B()
o3 = C()
print(obj1, obj2, obj3)
```

a) 1 1 1

b) 1 2 3

c) '1' '1' '1'

d) An exception is thrown

View Answer

Answer: b
Explanation: When different objects are invoked, each of the individual classes return their individual values and hence it is printed.

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

```
class A:
    def __init__(self):
        self.multiply(15)
        print(self.i)

    def multiply(self, i):
        self.i = 4 * i;
class B(A):
    def __init__(self):
        super().__init__()

    def multiply(self, i):
        self.i = 2 * i;
obj = B()
```

a) 15

b) 60

c) An exception is thrown

d) 30

View Answer

Answer: d
Explanation: The derived class B overrides base class A.

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

```
class Demo:
    def check(self):
        return " Demo's check "
    def display(self):
        print(self.check())
class Demo_Derived(Demo):
    def check(self):
        return " Derived's check "
Demo().display()
Demo_Derived().display()
```

a) Demo’s check Derived’s check

b) Demo’s check Demo’s check

c) Derived’s check Demo’s check

d) Syntax error

View Answer

Answer: a
Explanation: Demo().display() invokes the display() method in class Demo and Demo_Derived().display() invokes the display() method in class Demo_Derived.

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

```
class A:
    def __init__(self):
        self.multiply(15)
    def multiply(self, i):
        self.i = 4 * i;
class B(A):
    def __init__(self):
        super().__init__()
        print(self.i)

    def multiply(self, i):
        self.i = 2 * i;
obj = B()
```

a) 15

b) 30

c) An exception is thrown

d) 60

View Answer

Answer: b
Explanation: The derived class B overrides base class A.

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

```
class Demo:
    def __check(self):
        return " Demo's check "
    def display(self):
        print(self.check())
class Demo_Derived(Demo):
    def __check(self):
        return " Derived's check "
Demo().display()
Demo_Derived().display()
```

a) Demo’s check Derived’s check

b) Demo’s check Demo’s check

c) Derived’s check Demo’s check

d) Syntax error

View Answer

Answer: b
Explanation: The method check is private so it can’t be accessed by the derived class. Execute the code in the Python shell.

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

```
class A:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __str__(self):
        return 1
    def __eq__(self, other):
        return self.x * self.y == other.x * other.y
obj1 = A(5, 2)
obj2 = A(2, 5)
print(obj1 == obj2)
```

a) False

b) 1

c) True

d) An exception is thrown

View Answer

Answer: c
Explanation: Since 5*2==2*5, True is printed. Execute it in the Python shell to verify.

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

```
class A:
    def one(self):
        return self.two()
    def two(self):
        return 'A'
class B(A):
    def two(self):
```

```
        return 'B'
obj2=B()
print(obj2.two())
```

a) A

b) An exception is thrown

c) A B

d) B

View Answer

Answer: d
Explanation: The derived class method two() overrides the method two() in the base class A.

15. Which of the following statements is true?

a) A non-private method in a superclass can be overridden

b) A subclass method can be overridden by the superclass

c) A private method in a superclass can be overridden

d) Overriding isn't possible in Python

View Answer

Answer: a
Explanation: A public method in the base class can be overridden by the same named method in the subclass.

Python Questions and Answers – Encapsulation

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on "Encapsulation".

1. Which of these is not a fundamental features of OOP?

a) Encapsulation

b) Inheritance

c) Instantiation

d) Polymorphism

View Answer

Answer: c
Explanation: Instantiation simply refers to creation of an instance of class. It is not a fundamental feature of OOP.

2. Which of the following is the most suitable definition for encapsulation?

a) Ability of a class to derive members of another class as a part of its own definition

b) Means of bundling instance variables and methods in order to restrict access to certain class members

c) Focuses on variables and passing of variables to functions

d) Allows for implementation of elegant software that is well designed and easily

modified

View Answer

Answer: b
Explanation: The values assigned by the constructor to the class members is used to create the object.

3. What is the output of the following code?

```
class Demo:
    def __init__(self):
        self.a = 1
        self.__b = 1

    def display(self):
        return self.__b
obj = Demo()
print(obj.a)
```

a) The program has an error because there isn't any function to return self.a

b) The program has an error because b is private and display(self) is returning a private member

c) The program runs fine and 1 is printed

d) The program has an error as you can't name a class member using __b

View Answer

Answer: c
Explanation: The program has no error because the class member which is public is printed. 1 is displayed. Execute in python shell to verify.

4. What is the output of the following code?

```
class Demo:
    def __init__(self):
        self.a = 1
        self.__b = 1

    def display(self):
        return self.__b

obj = Demo()
print(obj.__b)
```

a) The program has an error because there isn't any function to return self.a

b) The program has an error because b is private and display(self) is returning a private member

c) The program has an error because b is private and hence can't be printed

d) The program runs fine and 1 is printed

View Answer

Answer: c
Explanation: Variables beginning with two underscores are said to be private members of the class and they can't be accessed directly.

5. Methods of a class that provide access to private members of the class are called as _____ and _____

a) getters/setters

b) __repr__/__str__

c) user-defined functions/in-built functions

d) __init__/__del__

View Answer

Answer: a
Explanation: The purpose of getters and setters is to get(return) and set(assign) private instance variables of a class.

6. Which of these is a private data field?

```
def Demo:
def __init__(self):
    __a = 1
    self.__b = 1
    self.__c__ = 1
    __d__= 1
```

a) __a

b) __b

c) __c__

d) __d__

View Answer

Answer: b
Explanation: Variables such as self.__b are private members of the class.

7. What is the output of the following code?

```
class Demo:
     def __init__(self):
         self.a = 1
         self.__b = 1

     def get(self):
         return self.__b

obj = Demo()
print(obj.get())
```

a) The program has an error because there isn't any function to return self.a

b) The program has an error because b is private and display(self) is returning a private member

c) The program has an error because b is private and hence can't be printed

d) The program runs fine and 1 is printed

View Answer

Answer: d
Explanation: Here, get(self) is a member of the class. Hence, it can even return a private member of the class. Because of this reason, the program runs fine and 1 is printed.

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

```
class Demo:
     def __init__(self):
         self.a = 1
         self.__b = 1
     def get(self):
         return self.__b
obj = Demo()
obj.a=45
print(obj.a)
```

a) The program runs properly and prints 45

b) The program has an error because the value of members of a class can't be changed from outside the class

c) The program runs properly and prints 1

d) The program has an error because the value of members outside a class can only be changed as self.a=45

View Answer

Answer: a
Explanation: It is possible to change the values of public class members using the object of the class.

9. Private members of a class cannot be accessed. True or False?

a) True

b) False

View Answer

Answer: b
Explanation: Private members of a class are accessible if written as follows: obj._Classname__privatemember. Such renaming of identifiers is called as name mangling.

10. The purpose of name mangling is to avoid unintentional access of private class members. True or False?

a) True

b) False

View Answer

Answer: a
Explanation: Name mangling prevents unintentional access of private members of a class, while still allowing access when needed. Unless the variable is accessed with its mangled name, it will not be found.

11. What is the output of the following code?

```
class fruits:
    def __init__(self):
        self.price = 100
        self.__bags = 5
    def display(self):
        print(self.__bags)
obj=fruits()
obj.display()
```

a) The program has an error because display() is trying to print a private class member

b) The program runs fine but nothing is printed

c) The program runs fine and 5 is printed

d) The program has an error because display() can't be accessed

View Answer

Answer: c
Explanation: Private class members can be printed by methods which are members of the class.

12. What is the output of the following code?

```
 class student:
    def __init__(self):
        self.marks = 97
        self.__cgpa = 8.7
    def display(self):
        print(self.marks)
obj=student()
print(obj._student__cgpa)
```

a) The program runs fine and 8.7 is printed

b) Error because private class members can't be accessed

c) Error because the proper syntax for name mangling hasn't been implemented

d) The program runs fine but nothing is printed

View Answer

Answer: a
Explanation: Name mangling has been properly implemented in the code given above and hence the program runs properly.

13. Which of the following is false about protected class members?

a) They begin with one underscore

b) They can be accessed by subclasses

c) They can be accessed by name mangling method

d) They can be accessed within a class

View Answer

Answer: c
Explanation: Protected class members can't be accessed by name mangling.

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

```
class objects:
```

```
    def __init__(self):
        self.colour = None
        self._shape = "Circle"

    def display(self, s):
        self._shape = s
obj=objects()
print(obj._objects_shape)
```

a) The program runs fine because name mangling has been properly implemented

b) Error because the member shape is a protected member

c) Error because the proper syntax for name mangling hasn't been implemented

d) Error because the member shape is a private member

View Answer

Answer: b
Explanation: Protected members begin with one underscore and they can only be accessed within a class or by subclasses.

Python Questions and Answers – Exception Handling – 1

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on "Exception Handling – 1".

1. How many except statements can a try-except block have?

a) zero

b) one

c) more than one

d) more than zero

View Answer

Answer: d
Explanation: There has to be at least one except statement.

2. When will the else part of try-except-else be executed?

a) always

b) when an exception occurs

c) when no exception occurs

d) when an exception occurs in to except block

View Answer

Answer: c
Explanation: The else part is executed when no exception occurs.

3. Is the following code valid?

```
try:
    # Do something
except:
```

```
    # Do something
finally:
    # Do something
```

a) no, there is no such thing as finally

b) no, finally cannot be used with except

c) no, finally must come before except

d) yes

View Answer

Answer: b
Explanation: Refer documentation.

4. Is the following code valid?

```
try:
    # Do something
except:
    # Do something
else:
    # Do something
```

a) no, there is no such thing as else

b) no, else cannot be used with except

c) no, else must come before except

d) yes

View Answer

Answer: d
Explanation: Refer documentation.

5. Can one block of except statements handle multiple exception?

a) yes, like except TypeError, SyntaxError [,…].

b) yes, like except [TypeError, SyntaxError].

c) no

d) none of the mentioned

View Answer

Answer: a
Explanation: Each type of exception can be specified directly. There is no need to put it in a list.

6. When is the finally block executed?

a) when there is no exception

b) when there is an exception

c) only if some condition that has been specified is satisfied

d) always

View Answer

Answer: d
Explanation: The finally block is always executed.

7. What is the output of the following code?

```
def foo():
    try:
        return 1
    finally:
        return 2
k = foo()
print(k)
```

a) 1

b) 2

c) 3

d) error, there is more than one return statement in a single try-finally block

View Answer

Answer: b
Explanation: The finally block is executed even there is a return statement in the try block.

8. What is the output of the following code?

```
def foo():
    try:
        print(1)
    finally:
        print(2)
foo()
```

a) 1 2

b) 1

c) 2

d) none of the mentioned

View Answer

Answer: a
Explanation: No error occurs in the try block so 1 is printed. Then the finally block is executed and 2 is printed.

9. What is the output of the following?

```
try:
    if '1' != 1:
        raise "someError"
    else:
        print("someError has not occured")
except "someError":
    print ("someError has occured")
```

a) someError has occured

b) someError has not occured

c) invalid code

d) none of the mentioned

View Answer

Answer: c
Explanation: A new exception class must inherit from a BaseException. There is no such inheritance here.

10. What happens when ‘1’ == 1 is executed?

a) we get a True

b) we get a False

c) an TypeError occurs

d) a ValueError occurs

View Answer

Answer: b
Explanation: It simply evaluates to False and does not raise any exception.

Python Questions and Answers – Exception Handling – 2

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Exception Handling – 2”.

1. The code shown below will result in an error if the input value is entered as -5. State whether this statement is true or false.

```
assert False, 'Spanish'
```

a) True

b) False

View Answer

Answer: a
Explanation: The code shown above results in an assertion error. The output of the code is:
Traceback (most recent call last):
File “ “, line 1, in
assert False, ‘Spanish’
AssertionError: Spanish
Hence, this statement is true.

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

```
x=10
y=8
assert x>y, 'X too small'
```

a) Assertion Error

b) 10 8

c) No output

d) 108

View Answer

Answer: c
Explanation: The code shown above results in an error if and only if xy, there is no error. Since there is no print statement, hence there is no output.

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

```
#generator
def f(x):
    yield x+1
g=f(8)
print(next(g))
```

a) 8

b) 9

c) 7

d) Error

View Answer

Answer: b
Explanation: The code shown above returns the value of the expression x+1, since we have used to keyword yield. The value of x is 8. Hence the output of the code is 9.

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

```
def f(x):
    yield x+1
    print("test")
    yield x+2
g=f(9)
```

a) Error

b) test

c) test

10

12

d) No output

View Answer

Answer: d
Explanation: The code shown above will not yield any output. This is because when we try to yield 9, and there is no next(g), the iteration stops. Hence there is no output.

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

```
def f(x):
    yield x+1
    print("test")
    yield x+2
g=f(10)
print(next(g))
print(next(g))
```

a) No output

b) 11

test

12

c) 11

test

d) 11

View Answer

Answer: b
Explanation: The code shown above results in the output:
11
test
12
This is because we have used next(g) twice. Had we not used next, there would be no output.

6. What is the output of the following code?

```
def a():
    try:
        f(x, 4)
    finally:
        print('after f')
    print('after f?')
a()
```

a) No output

b) after f?

c) error

d) after f

View Answer

Answer: c
Explanation: This code shown above will result in an error simply because ‘f’ is not defined. ‘try’ and ‘finally’ are keywords used in exception handling.

7. What is the output of the code shown?

```
def f(x):
    for i in range(5):
        yield i
g=f(8)
print(list(g))
```

a) [0, 1, 2, 3, 4]

b) [1, 2, 3, 4, 5, 6, 7, 8]

c) [1, 2, 3, 4, 5]

d) [0, 1, 2, 3, 4, 5, 6, 7]

View Answer

Answer: a
Explanation: The output of the code shown above is a list containing whole numbers in the range (5). Hence the output of this code is: [0, 1, 2, 3, 4].

8. The error displayed in the code shown below is:

```
import itertools
l1=(1, 2, 3)
l2=[4, 5, 6]
l=itertools.chain(l1, l2)
print(next(l1))
```

a) ‘list’ object is not iterator

b) ‘tuple’ object is not iterator

c) ‘list’ object is iterator

d) ‘tuple’ object is iterator

View Answer

Answer: b
Explanation: The error raised in the code shown above is that: ‘tuple’ object is not iterator. Had we given l2 as argument to next, the error would have been: ‘list’ object is not iterator.

9. Which of the following is not an exception handling keyword in Python?

a) try

b) except

c) accept

d) finally

View Answer

Answer: c
Explanation: The keywords ‘try’, ‘except’ and ‘finally’ are exception handling keywords in python whereas the word ‘accept’ is not a keyword at all.

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

```
g = (i for i in range(5))
type(g)
```

a) class <’loop’>

b) class <‘iteration’>

c) class <’range’>

d) class <’generator’>

View Answer

Answer: d
Explanation: Another way of creating a generator is to use parenthesis. Hence the output of the code shown above is: class<’generator’>.

Python Questions and Answers – Exception Handling – 3

This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Exception Handling – 3”.

1. What happens if the file is not found in the code shown below?

```
a=False
while not a:
    try:
        f_n = input("Enter file name")
        i_f = open(f_n, 'r')
    except:
        print("Input file not found")
```

a) No error

b) Assertion error

c) Input output error

d) Name error

View Answer

Answer: a
Explanation: In the code shown above, if the input file in not found, then the statement: “Input file not found” is printed on the screen. The user is then prompted to reenter the file name. Error is not thrown.

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

```
lst = [1, 2, 3]
lst[3]
```

a) NameError

b) ValueError

c) IndexError

d) TypeError

View Answer

Answer: c
Explanation: The snippet of code shown above throws an index error. This is because the index of the list given in the code, that is, 3 is out of range. The maximum index of this list is 2.

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

```
t[5]
```

a) IndexError

b) NameError

c) TypeError

d) ValeError

View Answer

Answer: b
Explanation: The expression shown above results in a name error. This is because the name ‘t’ is not defined.

4. What is the output of the following code, if the time module has already been imported?

```
4 + '3'
```

a) NameError

b) IndexError

c) ValueError

d) TypeError

View Answer

Answer: d
Explanation: The line of code shown above will result in a type error. This is because the operand '+' is not supported when we combine the data types 'int' and 'str'. Sine this is exactly what we have done in the code shown above, a type error is thrown.

5. The output of the code shown below is:

```
int('65.43')
```

a) ImportError

b) ValueError

c) TypeError

d) NameError

View Answer

Answer: b
Explanation: The snippet of code shown above results in a value error. This is because there is an invalid literal for int() with base 10: '65.43'.

6. Compare the two codes shown below and state the output if the input entered in each case is -6?

```
CODE 1
import math
num=int(input("Enter a number of whose factorial you want to find"))
print(math.factorial(num))

CODE 2
num=int(input("Enter a number of whose factorial you want to find"))
print(math.factorial(num))
```

a) ValueError, NameError

b) AttributeError, ValueError

c) NameError, TypeError

d) TypeError, ValueError

View Answer

Answer: a
Explanation: The first code results in a ValueError. This is because when we enter the input as -6, we are trying to find the factorial of a negative number, which is not possible. The second code results in a NameError. This is because we have not imported the math module. Hence the name 'math' is undefined.

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

```
def getMonth(m):
    if m<1 or m>12:
        raise ValueError("Invalid")
    print(m)
getMonth(6)
```

a) ValueError

b) Invalid

c) 6

d) ValueError(“Invalid”)

View Answer

Answer: c
Explanation: In the code shown above, since the value passed as an argument to the function is between 1 and 12 (both included), hence the output is the value itself, that is 6. If the value had been above 12 and less than 1, a ValueError would have been thrown.

8. What is the output of the code shown below if the input entered is 6?

```
valid = False
while not valid:
    try:
        n=int(input("Enter a number"))
        while n%2==0:
            print("Bye")
        valid = True
    except ValueError:
        print("Invalid")
```

a) Bye (printed once)

b) No output

c) Invalid (printed once)

d) Bye (printed infinite number of times)

View Answer

Answer: d
Explanation: The code shown above results in the word “Bye” being printed infinite number of times. This is because an even number has been given as input. If an odd number had been given as input, then there would have been no output.

9. Identify the type of error in the codes shown below.

```
Print(“Good Morning”)
print(“Good night)
```

a) Syntax, Syntax

b) Semantic, Syntax

c) Semantic, Semantic

d) Syntax, Semantic

View Answer

Answer: b
Explanation: The first code shows an error detected during execution. This might occur

occasionally. The second line of code represents a syntax error. When there is deviation from the rules of a language, a syntax error is thrown.

10. Which of the following statements is true?

a) The standard exceptions are automatically imported into Python programs

b) All raised standard exceptions must be handled in Python

c) When there is a deviation from the rules of a programming language, a semantic error is thrown

d) If any exception is thrown in try block, else block is executed

View Answer

Answer: a
Explanation: When any exception is thrown in try block, except block is executed. If exception in not thrown in try block, else block is executed. When there is a deviation from the rules of a programming language, a syntax error is thrown. The only true statement above is: The standard exceptions are automatically imported into Python programs.

11. Which of the following is not a standard exception in Python?

a) NameError

b) IOError

c) AssignmentError

d) ValueError

View Answer

Answer: c
Explanation: NameError, IOError and ValueError are standard exceptions in Python whereas Assignment error is not a standard exception in Python.

12. Syntax errors are also known as parsing errors. Is this statement true or false?

a) True

b) False

View Answer

Answer: a
Explanation: Syntax errors are known as parsing errors. Syntax errors are raised when there is a deviation from the rules of a language. Hence the statement is true.

13. An exception is:

a) an object

b) a special function

c) a standard module

d) a module

View Answer

Answer: a
Explanation: An exception is an object that is raised by a function signaling that an unexpected situation has occurred, that the function itself cannot handle.

14. _____ exceptions are raised as a result of an error in opening a particular file.

a) ValueError

b) TypeError

c) ImportError

d) IOError

View Answer

Answer: d
Explanation: IOError exceptions are raised as a result of an error in opening or closing a particular file.

15. Which of the following blocks will be executed whether an exception is thrown or not?

a) except

b) else

c) finally

d) assert

View Answer

Answer: c
Explanation: The statements in the finally block will always be executed, whether an exception is thrown or not. This clause is used to close the resources used in a code.