**Python Question and Answers – Built-in Functions – 1**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Built-in Functions – 1”.**

1. Which of the following functions is a built-in function in python?

a) seed() b) sqrt() c) factorial() d) print()

**Answer**: d

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

2. What is the output of the expression:

round(4.576)

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

**Answer**: b

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

3. The function pow(x,y,z) is evaluated as:

a) (x\*\*y)\*\*z b) (x\*\*y) / z

c) (x\*\*y) % z d) (x\*\*y)\*z

**Answer**: c

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

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

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

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

**Answer**: c

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

5. What is the output of the expression?

round(4.5676,2)?

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

**Answer**: c

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

6. What is the output of the following function?

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

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

**Answer**: b

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

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

import math

abs(math.sqrt(25))

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

**Answer**: d

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

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

sum(2,4,6)

sum([1,2,3])

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

**Answer**: a

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

9. What is the output of the function:

all(3,0,4.2)

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

**Answer**: c

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

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

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

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

**Answer**: b

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

**Python Question and Answers – Built-in Functions – 2**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Built-in Functions – 2”.**

1. What are the outcomes of the following functions?

chr(‘97’)

chr(97)

a) a and Erro b) ‘a’ c) Error d) Error

Error

**Answer**: c

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

2. What is the output of the following function?

complex(1+2j)

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

**Answer**: d

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

3. What is the output of the function complex() ?

a) 0j b) 0+0j c) 0 d) Error

**Answer**: a

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

4. The function divmod(a,b), where both ‘a’ and ‘b’ are integers is evaluated as:

a) (a%b, a//b) b) (a//b, a%b)

c) (a//b, a\*b) c) (a/b, a%b)

**Answer**: b

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

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

divmod(10.5,5)

divmod(2.4,1.2)

a) (2.00, 0.50) and (2.00, 0.00) b) (2, 0.5) and (2, 0)

c) (2.0, 0.5) and (2.0, 0.0) d) (2, 0.5) and (2)

**Answer**: c

***Explanation:***  See python documentation for the function divmod.

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

a) True b) False

**Answer**: a

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

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

list(enumerate([2, 3]))

a) Error b) [(1, 2), (2, 3)]

c) [(0, 2), (1, 3)] d) [(2, 3)]

**Answer**: c

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

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

x=3

eval('x^2')

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

**Answer**: b

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

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

float('1e-003')

float('2e+003')

a) 3.00 and 300 b) 0.001 and 2000.0

c) 0.001 and 200 d) Error and 2003

**Answer**: b

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

10. Which of the following functions does not necessarily accept only iterables as arguments?

a) enumerate() b) all() c) chr() d) max()

**Answer**: c

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

**Python Question and Answers – Built-in Functions – 3**

**This set of Python Multiple Choice Questions & Answers (MCQs) focuses on “Built-in Functions – 3”.**

1. Which of the following functions accepts only integers as arguments?

a) ord() b) min() c) chr() d) any()

**Answer**: c

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

2. Suppose there is a list such that: l=[2,3,4].

If we want to print this list in reverse order, which of the following methods should be used?

a) reverse(l) b) list(reverse[(l)])

c) reversed(l) d) list(reversed(l))

**Answer**: d

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

3. The output of the function:

float(' -12345\n')

(Note that the number of blank spaces before the number is 5)

a) -12345.0 (5 blank spaces before the number)

b) -12345.0

c) Error

d) -12345.000000000…. (infinite decimal places)

**Answer**: b

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

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

ord(65)

ord(‘A’)

a) A and 65 b) Error and 65

c) A and Error c) Error and Error

**Answer**: b

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

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

float(‘-infinity’)

float(‘inf’)

a) –inf and inf b) –infinity and inf

c) Error and Error d) Error and Junk value

**Answer**: a

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

6. Which of the following functions will not result in an error when no arguments are passed to it?

a) min() b) divmod() c) all() d) float()

**Answer**: d

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

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

hex(15)

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

**Answer**: d

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

8. Which of the following functions does not throw an error?

a) ord() b) ord(‘ ‘) c) ord(”) d) ord(“”)

**Answer**: b

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

9. What is the output of the function:

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

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

**Answer**: a

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

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

oct(7)

oct(‘7’)

a) Error and 07 b) 0o7 and Error

c) 0o7 and Error d) 07 and 0o7

**Answer**: c

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

**Python MCQ of – Function – 1**

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

1. Which of the following is the use of function in python?

a) Functions are reusable pieces of programs

b) Functions don’t provide better modularity for your application

c) you can’t also create your own functions

d) All of the mentioned

**Answer**: a

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

2. Which keyword is use for function?

a) Fun b) Define c) Def d) Function

**Answer**: c

***Explanation:***  None.

3. What is the output of the below program?

def sayHello():

print('Hello World!')

sayHello()

sayHello()

a) Hello World! and Hello World!

b) ‘Hello World!’ and ‘Hello World!’

c) Hello and Hello

d) None of the mentioned

**Answer**: a

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

def sayHello():

print('Hello World!') # block belonging to the function

# End of function #

sayHello() # call the function

sayHello() # call the function again

4. What is the output of the below program?

def printMax(a, b):

if a > b:

print(a, 'is maximum')

elif a == b:

print(a, 'is equal to', b)

else:

print(b, 'is maximum')

printMax(3, 4)

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

**Answer**: c

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

5. What is the output of the below program ?

x = 50

def func(x):

print('x is', x)

x = 2

print('Changed local x to', x)

func(x)

print('x is now', x)

a) x is now 50 b) x is now 2

c) x is now 100 d) None of the mentioned

**Answer**: a

***Explanation:***  The first time that we print the value of the name x with the first line in the function’s body, Python uses the value of the parameter declared in the main block, above the function definition.

Next, we assign the value 2 to x. The name x is local to our function. So, when we change the value of x in the function, the x defined in the main block remains unaffected.

With the last print function call, we display the value of x as defined in the main block, thereby confirming that it is actually unaffected by the local assignment within the previously called function.

6. What is the output of the below program?

x = 50

def func():

global x

print('x is', x)

x = 2

print('Changed global x to', x)

func()

print('Value of x is', x)

a) x is 50

Changed global x to 2

Value of x is 50

b) x is 50

Changed global x to 2

Value of x is 2

c) x is 50

Changed global x to 50

Value of x is 50

d) None of the mentioned

**Answer**: b

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

7. What is the output of below program?

def say(message, times = 1):

print(message \* times)

say('Hello')

say('World', 5)

a) Hello and WorldWorldWorldWorldWorld

b) Hello and World 5

c) Hello and World,World,World,World,World

d) Hello and HelloHelloHelloHelloHello

**Answer**: a

***Explanation:***  For some functions, you may want to make some parameters optional and use default values in case the user does not want to provide values for them. This is done with the help of default argument values. You can specify default argument values for parameters by appending to the parameter name in the function definition the assignment operator (=) followed by the default value.

The function named say is used to print a string as many times as specified. If we don’t supply a value, then by default, the string is printed just once. We achieve this by specifying a default argument value of 1 to the parameter times.

In the first usage of say, we supply only the string and it prints the string once. In the second usage of say, we supply both the string and an argument 5 stating that we want to say the string message 5 times.

8. What is the output of the below program?

def func(a, b=5, c=10):

print('a is', a, 'and b is', b, 'and c is', c)

func(3, 7)

func(25, c = 24)

func(c = 50, a = 100)

a) a is 7 and b is 3 and c is 10

a is 25 and b is 5 and c is 24

a is 5 and b is 100 and c is 50

b) a is 3 and b is 7 and c is 10

a is 5 and b is 25 and c is 24

a is 50 and b is 100 and c is 5

c) a is 3 and b is 7 and c is 10

a is 25 and b is 5 and c is 24

a is 100 and b is 5 and c is 50

d) None of the mentioned

**Answer**: c

***Explanation:***  If you have some functions with many parameters and you want to specify only some of them, then you can give values for such parameters by naming them – this is called keyword arguments – we use the name (keyword) instead of the position (which we have been using all along) to specify the arguments to the function.

The function named func has one parameter without a default argument value, followed by two parameters with default argument values.

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

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

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

9. What is the output of below program?

def maximum(x, y):

if x > y:

return x

elif x == y:

return 'The numbers are equal'

else:

return y

print(maximum(2, 3))

a) 2 b) 3 c) The numbers are equal

d) None of the mentioned

**Answer**: b

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

10. Which of the following is a features of DocString?

a) Provide a convenient way of associating documentation with Python modules, functions, classes, and methods

b) All functions should have a docstring

c) Docstrings can be accessed by the \_\_doc\_\_ attribute on objects

d) All of the mentioned

**Answer**: d

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

**Python MCQ of – Function – 2**

**This set of Python Questions for entrance examinations focuses on “Functions”.**

1. Which are the advantages of functions in python?

a) Reducing duplication of code

b) Decomposing complex problems into simpler pieces

c) Improving clarity of the code

d) All of the mentioned

**Answer**: d

***Explanation:***  None.

2. What are the two main types of functions?

a) Custom function

b) Built-in function & User defined function

c) User function

d) System function

**Answer**: b

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

3. Where is function defined?

a) Module b) Class

c) Another function d) All of the mentioned

**Answer**: d

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

4. What is called when a function is defined inside a class?

a) Module b) Class

c) Another function d) Method

**Answer**: d

***Explanation:***  None.

5. Which of the following is the use of id() function in python?

a) Id returns the identity of the object

b) Every object doesn’t have a unique id

c) All of the mentioned

d) None of the mentioned

**Answer**: a

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

6. Which of the following refers to mathematical function?

a) sqrt b) rhombus c) add d) rhombus

**Answer**: a

***Explanation:***  Functions that are always available for usage, functions that are contained within external modules, which must be imported and functions defined by a programmer with the def keyword.

Eg: math import sqrt

A sqrt() function is imported from the math module.

7. What is the output of below program?

def cube(x):

return x \* x \* x

x = cube(3)

print x

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

**Answer**: c

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

8. What is the output of the below program?

def C2F(c):

return c \* 9/5 + 32

print C2F(100)

print C2F(0)

a) 212 and 32 b) 314 and 24

c) 567 and 98 d) None of the mentioned

**Answer**: a

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

9. What is the output of the below program?

def power(x, y=2):

r = 1

for i in range(y):

r = r \* x

return r

print power(3)

print power(3, 3)

a) 212 and 32 b) 9 and 27

c) 567 and 98 d) None of the mentioned

**Answer**: b

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

10. What is the output of the below program?

def sum(\*args):

'''Function returns the sum

of all values'''

r = 0

for i in args:

r += i

return r

print sum.\_\_doc\_\_

print sum(1, 2, 3)

print sum(1, 2, 3, 4, 5)

a) 6 and 15 b) 6 and 100

c) 123 and 12345 d) None of the mentioned

**Answer**: a

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

**Python MCQ of – Function – 3**

**This set of Python Questions for campus interview focuses on “Functions”.**

1. Python supports the creation of anonymous functions at runtime, using a construct called \_\_\_\_\_\_\_\_\_\_

a) Lambda

b) pi

c) anonymous

d) None of the mentioned

**Answer**: a

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

2. What is the output of this program?

y = 6

z = lambda x: x \* y

print z(8)

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

**Answer**: a

***Explanation:***  The lambda keyword creates an anonymous function. The x is a parameter, that is passed to the lambda function. The parameter is followed by a colon character. The code next to the colon is the expression that is executed, when the lambda function is called. The lambda function is assigned to the z variable.

The lambda function is executed. The number 8 is passed to the anonymous function and it returns 48 as the result. Note that z is not a name for this function. It is only a variable to which the anonymous function was assigned.

3. What is the output of below program?

lamb = lambda x: x \*\* 3

print(lamb(5))

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

**Answer**: c

***Explanation:***  None.

4. Does Lambda contains return statements?

a) True b) False

**Answer**: b

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

5. Lambda is a statement.

a) True b) False

6. Lambda contains block of statements

a) True b) False

**Answer**: b

***Explanation:***  None.

7. What is the output of below program?

def f(x, y, z): return x + y + z

f(2, 30, 400)

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

**Answer**: a

***Explanation:***  None.

8. What is the output of below program?

def writer():

title = 'Sir'

name = (lambda x:title + ' ' + x)

return name

who = writer()

who('Arthur')

a) Arthur Sir b) Sir Arthur

c) Arthur d) None of the mentioned

**Answer**: b

***Explanation:***  None.

9. What is the output of this program?

L = [lambda x: x \*\* 2,

lambda x: x \*\* 3,

lambda x: x \*\* 4]

for f in L:

print(f(3))

a) 27 and 81 and 343 b) 6 and 12

c) 9 and 27 and 81 d) None of the mentioned

**Answer**: c

***Explanation:***  None.

10. What is the output of this program?

min = (lambda x, y: x if x < y else y)

min(101\*99, 102\*98)

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

**Answer**: c

***Explanation:***  None.

**Python MCQ of – Function – 4**

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

1. What is a variable defined outside a function referred to as?

a)A static variable b)A global variable

c)A local variable d)An automatic variable

**Answer**: b

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

2.What is a variable defined inside a function referred to as?

a)A global variable b)A volatile variable

c)A local variable d)An automatic variable

**Answer**: c

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

3.What is the output of the following code?

i=0

def change(i):

i=i+1

return i

change(1)

print(i)

a)1

b)Nothing is displayed

c)0

d)An exception is thrown

**Answer**: c

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

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

def a(b):

b = b + [5]

c = [1, 2, 3, 4]

a(c)

print(len(c))

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

**Answer**: b

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

5.What is the output of the following code?

a=10

b=20

def change():

global b

a=45

b=56

change()

print(a)

print(b)

a)10 and 56 b)45 and 56

c)10 and 20 d)Syntax Error

**Answer**: a

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

6. What is the output of the following code?

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

i = i + j

j = j + 1

print(i, j)

change(j = 1, i = 2)

a)An exception is thrown because of conflicting values

b)1 2 c)3 3 d)3 2

**Answer**: d

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

7.What is the output of the following code?

def change(one, \*two):

print(type(two))

change(1,2,3,4)

a)Integer b)Tuple

c)Dictionary d)An exception is thrown

**Answer**: b

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

8. If a function doesn’t have a return statement, which of the following does the function return?

a)int b)null c)None

d)An exception is thrown without the return statement

**Answer**: c

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

9. What is the output of the following code?

def display(b, n):

while n > 0:

print(b,end="")

n=n-1

display('z',3)

a)zzz b)zz

c)An exception is executed d)Infinite loop

**Answer**: a

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

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

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

print(type(b))

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

a)String b)Tuple

c)Dictionary d)An exception is thrown

**Answer**: c

***Explanation:***  b combines the remaining parameters into a dictionary.

**Python MCQ of – Argument Parsing 1**

**This set of Python Puzzles focuses on “Argument Parsing”.**

1. What is the type of each element in sys.argv?

a) set b) list c) tuple d) string

**Answer**: d

***Explanation:***  It is a list of strings.

2. What is the length of sys.argv?

a) number of arguments b) number of arguments + 1

c) number of arguments – 1 d) none of the mentioned

**Answer**: b

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

3. What is the output of the following code?

def foo(k):

k[0] = 1

q = [0]

foo(q)

print(q)

a) [0]. ) [1]. c) [1, 0]. d) [0, 1].

**Answer**: b

***Explanation:***  Lists are passed by reference.

4. How are keyword arguments specified in the function heading?

a) one star followed by a valid identifier

b) one underscore followed by a valid identifier

c) two stars followed by a valid identifier

d) two underscores followed by a valid identifier

**Answer**: c

***Explanation:***  Refer documentation.

5. How many keyword arguments can be passed to a function in a single function call?

a) zero b) one c) zero or more d) one or more

**Answer**: c

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

6. What is the output of the following code?

def foo(fname, val):

print(fname(val))

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

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

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

**Answer**: a

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

7. What is the output of the following code?

def foo():

return total + 1

total = 0

print(foo())

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

**Answer**: b

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

8. What is the output of the following code?

def foo():

total += 1

return total

total = 0

print(foo())

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

**Answer**: c

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

9. What is the output of the following code?

def foo(x):

x = ['def', 'abc']

return id(x)

q = ['abc', 'def']

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

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

**Answer**: b

***Explanation:***  A new object is created in the function.

10. What is the output of the following code?

def foo(i, x=[]):

x.append(i)

return x

for i in range(3):

print(foo(i))

a) [0] [1] [2] b) [0] [0, 1] [0, 1, 2].

c) [1] [2] [3]. d) [1] [1, 2] [1, 2, 3].

**Answer**: b

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

**Python MCQ of – Argument Parsing 2**

**This set of Tricky Python Questions & Answers focuses on “Argument Parsing”.**

1. What is the output of the following code?

def foo(k):

k = [1]

q = [0]

foo(q)

print(q)

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

**Answer**: a

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

2. How are variable length arguments specified in the function heading?

a) one star followed by a valid identifier

b) one underscore followed by a valid identifier

c) two stars followed by a valid identifier

d) two underscores followed by a valid identifier

**Answer**: a

***Explanation:***  Refer documentation.

3. Which module in the python standard library parses options received from the command line?

a) getopt b) os c) getarg d) main

**Answer**: a

***Explanation:***  getopt parses options received from the command line.

4. What is the type of sys.argv?

a) set b) list c) tuple d) string

**Answer**: b

***Explanation:***  It is a list of elements.

5. What is the value stored in sys.argv[0]?

a) null

b) you cannot access it

c) the program’s name

d) the first argument

**Answer**: c

***Explanation:***  Refer documentation.

6. How are default arguments specified in the function heading?

a) identifier followed by an equal to sign and the default value

b) identifier followed by the default value within backticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

**Answer**: a

***Explanation:***  Refer documentation.

7. How are required arguments specified in the function heading?

a) identifier followed by an equal to sign and the default value

b) identifier followed by the default value within backticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

**Answer**: d

***Explanation:***  Refer documentation.

8. What is the output of the following code?

def foo(x):

x[0] = ['def']

x[1] = ['abc']

return id(x)

q = ['abc', 'def']

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

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

**Answer**: a

***Explanation:***  The same object is modified in the function.

9. Where are the arguments received from the command line stored?

a) sys.argv b) os.argv

c) argv d) none of the mentioned

**Answer**: a

***Explanation:***  Refer documentation.

10. What is the output of the following?

def foo(i, x=[]):

x.append(x.append(i))

return x

for i in range(3):

y = foo(i)

print(y)

a) [[[0]], [[[0]], [1]], [[[0]], [[[0]], [1]], [2]]].

b) [[0], [[0], 1], [[0], [[0], 1], 2]].

c) [0, None, 1, None, 2, None].

d) [[[0]], [[[0]], [1]], [[[0]], [[[0]], [1]], [2]]].

**Answer**: c

***Explanation:***  append() returns None.

**Python MCQ of – Global vs Local Variables – 1**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Global vs Local Variables – 1”.**

1. The output of the code shown below is:

def f1():

x=15

print(x)

x=12

f1()

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

**Answer**: c

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

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

def f1():

x=100

print(x)

x=+1

f1()

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

**Answer**: b

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

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

def san(x):

print(x+1)

x=-2

x=4

san(12)

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

**Answer**: a

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

4. What is the output of the code shown?

def f1():

global x

x+=1

print(x)

x=12

print("x")

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

**Answer**: d

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

13

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

def f1(x):

global x

x+=1

print(x)

f1(15)

print("hello")

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

hello

**Answer**: a

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

hello

6. What is the output of the following code?

x=12

def f1(a,b=x):

print(a,b)

x=15

f1(4)

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

**Answer**: c

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

7. What is the output of the code shown?

def f():

global a

print(a)

a = "hello"

print(a)

a = "world"

f()

print(a)

a) hello **and** hello **and** world

b) world **and** world **and** hello

c) hello **and** world **and**world

d) world **and** hello **and** world

**Answer**: b

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

world

hello

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

def f1(a,b=[]):

b.append(a)

return b

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

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

**Answer**: d

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

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

def f(p, q, r):

global s

p = 10

q = 20

r = 30

s = 40

print(p,q,r,s)

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

f(5,10,15)

a) 1 2 3 4 b) 5 10 15 4

c) 10 20 30 40 d) 5 10 15 40

**Answer**: c

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

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

def f(x):

print("outer")

def f1(a):

print("inner")

print(a,x)

f(3)

f1(1)

a) outer and error

b) inner and error

c) outer and inner

d) error

**Answer**: a

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

11. The output of code shown below is:

x = 5

def f1():

global x

x = 4

def f2(a,b):

global x

return a+b+x

f1()

total = f2(1,2)

print(total)

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

**Answer**: b

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

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

x=100

def f1():

global x

x=90

def f2():

global x

x=80

print(x)

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

**Answer**: a

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

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

y, z = 1, 2

def f():

global x

x = y+z

a) x b) y and z

c) x, y and z d) Neither x, nor y, nor z

**Answer**: c

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

**Python MCQ of – 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

**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()

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

**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

**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

**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

**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

**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()

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

**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() and globals() b) locals() and locals()

c) globals() and locals() d) globals() and globals()

**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 MCQ of – 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

**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

**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

**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)

**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

**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

**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

**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

**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

**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

**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

**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

**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

**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

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

**Answer**: d

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

**Python MCQ of – 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

**Answer**: a

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

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

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

l2=l.copy()

l2[0]=88

l2

a) [88, 2, 3, [4, 5]] and [88, 2, 3, [4, 5]]

b) [2, 3, [4, 5]] and [88, 2, 3, [4, 5]]

c) [88, 2, 3, [4, 5]] and [2, 3, [4, 5]]

d) [2, 3, [4, 5]] and [2, 3, [4, 5]]

**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

**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

**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

**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

**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

**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]] and [10, 20, 30, 90]

b) Error

c) [10, 20, 30 [90]] and [10, 20, 30, [40]]

d) [10, 20, 30, [40]] and [10, 20, 30, [90]]

**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

**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)] and [1, 2, 3, 4]

b) [1, 2, 3, 4] and [1, 2, 3, (4)]

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

d) [1, 2, 3, (4)] and [1, 2, 3, (4)]

**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

**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 Merge Sort algorithm when it is solved recursively?

a)n=0 b)n=1 c)A list of length one d)An empty list

**Answer**: c

***Explanation:***  Merge Sort 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.

**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 MCQ of – 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

**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

**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

**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]

**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

**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]

**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

**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

**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 **and** Error b) False False

c) True False **and** Error d) False True

**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

**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

**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)

**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

**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

**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

**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 MCQ of – 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

**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

**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

**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]

**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 **and** 16

**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]

**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

**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

**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 **and** Error b) False False

c) True False **and** Error d) False True

**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

**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

**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)

**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

**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

**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

**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 MCQ of – Mapping Functions – 1**

**This set of Python Interview Questions & Answers focuses on “Mapping Functions”.**

1. What is the output of the following?

elements = [0, 1, 2]

def incr(x):

return x+1

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

a) [1, 2, 3]. b) [0, 1, 2].

c) error d) none of the mentioned

**Answer**: c

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

2. What is the output of the following?

elements = [0, 1, 2]

def incr(x):

return x+1

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

a) [1, 2, 3]. b) [0, 1, 2].

c) error d) none of the mentioned

**Answer**: a

***Explanation:***  Each element of the list is incremented.

3. What is the output of the following?

x = ['ab', 'cd']

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

a) [‘AB’, ‘CD’]. b) [‘ab’, ‘cd’].

c) error d) none of the mentioned

**Answer**: c

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

4. What is the output of the following?

def to\_upper(k):

return k.upper()

x = ['ab', 'cd']

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

a) [‘AB’, ‘CD’]. b) [‘ab’, ‘cd’].

c) none of the mentioned d) error

**Answer**: d

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

5. What is the output of the following?

def to\_upper(k):

return k.upper()

x = ['ab', 'cd']

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

a) [‘AB’, ‘CD’]. b) [‘ab’, ‘cd’].

c) none of the mentioned d) error

**Answer**: a

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

6. What is the output of the following?

def to\_upper(k):

k.upper()

x = ['ab', 'cd']

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

a) [‘AB’, ‘CD’]. b) [‘ab’, ‘cd’].

c) none of the mentioned d) error

**Answer**: c

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

7. What is the output of the following?

x = ['ab', 'cd']

print(map(len, x))

a) [‘ab’, ‘cd’].

b) [2, 2].

c) [‘2’, ‘2’]. d) none of the mentioned

**Answer**: d

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

8. What is the output of the following?

x = ['ab', 'cd']

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

a) [‘ab’, ‘cd’]. b) [2, 2].

c) [‘2’, ‘2’]. d) none of the mentioned

**Answer**: b

***Explanation:***  The length of each string is 2.

9. What is the output of the following?

x = ['ab', 'cd']

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

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

**Answer**: d

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

10. What is the output of the following?

x = ['ab', 'cd']

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

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

**Answer**: a

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

**Python MCQ of – Mapping Functions – 2**

**This set of Tough Python Interview Questions & Answers focuses on “Mapping Functions”.**

1. What is the output of the following?

x = ['ab', 'cd']

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

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

**Answer**: c

***Explanation:***  SyntaxError, unbalanced parenthesis.

2. What is the output of the following?

x = ['ab', 'cd']

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

a) [‘a’, ‘b’, ‘c’, ‘d’].

b) [[‘ab’], [‘cd’]].

c) [[‘a’, ‘b’], [‘c’, ‘d’]].

d) none of the mentioned

**Answer**: c

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

3. What is the output of the following?

x = [12, 34]

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

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

**Answer**: c

***Explanation:***  TypeError, int has no len().

4. What is the output of the following?

x = [12, 34]

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

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

**Answer**: a

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

5. What is the output of the following?

x = [12, 34]

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

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

**Answer**: c

***Explanation:***  Cannot perform join on a list of ints.

6. What is the output of the following?

x = [12, 34]

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

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

**Answer**: a

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

7. What is the output of the following?

x = [12, 34]

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

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

**Answer**: d

***Explanation:***  TypeError. Execute in shell to verify.

8. What is the output of the following?

x = [12.1, 34.0]

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

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

**Answer**: c

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

9. What is the output of the following?

x = [12.1, 34.0]

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

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

**Answer**: d

***Explanation:***  str(ab.c) is ‘ab.c’.

10. What is the output of the following?

x = [[0], [1]]

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

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

**Answer**: c

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

**Python MCQ of – Mapping Functions – 3**

**This set of Python Interview Questions and Answers for freshers focuses on**

**“Mapping Functions”.**

1. What is the output of the following?

x = [[0], [1]]

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

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

**Answer**: c

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

2. What is the output of the following?

x = [[0], [1]]

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

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

**Answer**: a

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

3. What is the output of the following?

x = [34, 56]

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

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

**Answer**: d

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

4. What is the output of the following?

x = [34, 56]

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

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

**Answer**: a

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

5. What is the output of the following?

x = [34, 56]

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

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

**Answer**: d

***Explanation:***  TypeError, map has no len.

6. What is the output of the following?

x = 'abcd'

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

a) [‘a’, ‘b’, ‘c’, ‘d’]. b) [‘abcd’].

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

**Answer**: c

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

7. What is the output of the following?

x = abcd

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

a) [‘a’, ‘b’, ‘c’, ‘d’]. b) [‘abcd’].

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

**Answer**: d

***Explanation:***  NameError, we have not defined abcd.

8. What is the output of the following?

x = 1234

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

a) [1, 2, 3, 4]. b) [1234].

c) [[1], [2], [3], [4]]. d) none of the mentioned

**Answer**: d

***Explanation:***  TypeError, int is not iterable.

9. What is the output of the following?

x = 1234

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

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

b) [1234].

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

d) none of the mentioned

**Answer**: d

***Explanation:***  TypeError, int is not iterable.

10. What is the output of the following?

x = 'abcd'

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

a) [‘a’, ‘b’, ‘c’, ‘d’]. b) [‘abcd’].

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

**Answer**: d

***Explanation:***  TypeError, list object is not callable.

11. Is Python code compiled or interpreted?

a)Python code is only compiled

b)Python code is both compiled and interpreted

c)Python code is only interpreted

d)Python code is neither compiled nor interpreted

**Answer**: b

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

12. Which of these is the definition for packages in Python?

a)A folder of python modules

b)A set of programs making use of Python modules

c)A set of main modules

d)A number of files containing Python definitions and statements

**Answer**: a

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

13. Which of these is false about a package?

a)A package can have subfolders and modules

b)Each import package need not introduce a namespace

c)import folder.subfolder.mod1 imports packages

d)from folder.subfolder.mod1 import objects imports packages

**Answer**: b

***Explanation:***  Packages provide a way of structuring Python namespace. Each import package introduces a namespace.

**Python MCQ of – Python Modules**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Python Modules”.**

1.Which of these definitions correctly describes a module?

a)Denoted by triple quotes for providing the specification of certain program elements

b)Design and implementation of specific functionality to be incorporated into a program

c)Defines the specification of how it is to be used

d)Any program that reuses code

**Answer**: b

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

2.Which of the following is not an advantage of using modules?

a)Provides a means of reuse of program code

b)Provides a means of dividing up tasks

c)Provides a means of reducing the size of the program

d)Provides a means of testing individual parts of the program

**Answer**: c

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

3.Program code making use of a given module is called a \_\_\_\_\_\_ of the module.

a)Client b)Docstring

c)Interface d)Modularity

**Answer**: a

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

4.\_\_\_\_\_\_ is a string literal denoted by triple quotes for providing the specifications of certain program elements.

a)Interface b)Modularity c)Client d)Docstring

**Answer**: d

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

5.Which of the following is true about top-down design process?

a)The details of a program design are addressed before the overall design

b)Only the details of the program are addressed

c)The overall design of the program is addressed before the details

d)Only the design of the program is addressed

**Answer**: c

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

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

a)True b)False

**Answer**: b

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

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

a)True b)False

**Answer**: b

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

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

#mod1

def change(a):

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

print(b)

#mod2

def change(a):

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

print(b)

from mod1 import change

from mod2 import change

#main

s=[1,2,3]

change(s)

a)[2,4,6]. b)[1,4,9].

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

**Answer**: d

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

9. Which of the following isn’t true about main modules?

a)When a python file is directly executed, it is considered main module of a program

b)Main modules may import any number of modules

c)Special name given to main modules is: \_\_main\_\_

d)Other main modules can import main modules

**Answer**: d

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

10. Which of the following is not a valid namespace?

a)Global namespace b)Public namespace

c)Built-in namespace d)Local namespace

**Answer**: b

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

11. Which of the following is false about “import modulename” form of import?

a)The namespace of imported module becomes part of importing module

b)This form of import prevents name clash

c)The namespace of imported module becomes available to importing module

d)The identifiers in module are accessed as: modulename.identifier

**Answer**: a

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

12. Which of the following is false about “from-import” form of import?

a)The syntax is: from modulename import identifier

b)This form of import prevents name clash

c)The namespace of imported module becomes part of importing module

d)The identifiers in module are accessed directly as: identifier

**Answer**: b

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

13. Which of the statements about modules is false?

a)In the “from-import” form of import, identifiers beginning with two underscores are private and aren’t imported

b)dir() built-in function monitors the items in the namespace of the main module

c)In the “from-import” form of import, all identifiers regardless of whether they are private or public are imported

d)When a module is loaded, a compiled version of the module with file extension .pyc is automatically produced

**Answer**: c

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

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

from math import factorial

print(math.factorial(5))

a)120

b)Nothing is printed

c)Error, method factorial doesn’t exist in math module

d)Error, the statement should be: print(factorial(5))

**Answer**: d

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

15. What is the order of namespaces in which Python looks for an identifier?

a)Python first searches the global namespace, then the local namespace and finally the built-in namespace

b)Python first searches the local namespace, then the global namespace and finally the built-in namespace

c)Python first searches the built-in namespace, then the global namespace and finally the local namespace

d)Python first searches the built-in namespace, then the local namespace and finally the global namespace

**Answer**: b

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

**Python MCQ of – Math – 1**

**This set of Python Objective Questions & Answers focuses on “Math – 1”.**

1. What is returned by math.ceil(3.4)?

a) 3 b) 4 c) 4.0 d) 3.0

**Answer**: b

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

2. What is the value returned by math.floor(3.4)?

a) 3 b) 4 c) 4.0 d) 3.0

**Answer**: a

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

3. What is the output of print(math.copysign(3, -1))?

a) 1 b) 1.0 c) -3 d) -3.0

**Answer**: d

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

4. What is displayed on executing print(math.fabs(-3.4))?

a) -3.4 b) 3.4 c) 3 d) -3

**Answer**: b

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

5. Is the output of the function abs() the same as that of the function math.fabs()?

a) sometimes b) always

c) never d) none of the mentioned

**Answer**: a

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

6. What is the value returned by math.fact(6)?

a) 720 b) 6 c) [1, 2, 3, 6]. d) error

**Answer**: d

***Explanation:***  NameError, fact() is not defined.

7. What is the value of x if x = math.factorial(0)?

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

**Answer**: b

***Explanation:***  Factorial of 0 is 1.

8. What is math.factorial(4.0)?

a) 24 b) 1 c) error d) none of the mentioned

**Answer**: a

***Explanation:***  The factorial of 4 is returned.

9. What is the output of print(math.factorial(4.5))?

a) 24 b) 120 c) error d) 24.0

**Answer**: c

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

10. What is math.floor(0o10)?

a) 8 b) 10 c) 0 d) 9

**Answer**: a

***Explanation:***  0o10 is 8 and floor(8) is 8.

**Python MCQ of – Math – 2**

**This set of Python Developer Questions & Answers focuses on “Mathematical Functions”.**

1. What does the function math.frexp(x) return?

a) a tuple containing of the mantissa and the exponent of x

b) a list containing of the mantissa and the exponent of x

c) a tuple containing of the mantissa of x

d) a list containing of the exponent of x

**Answer**: a

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

2. What is the result of math.fsum([.1 for i in range(20)])?

a) 2.0 b) 20 c) 2 d) 2.0000000000000004

**Answer**: a

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

3. What is the result of sum([.1 for i in range(20)])?

a) 2.0 b) 20 c) 2 d) 2.0000000000000004

**Answer**: d

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

4. What is returned by math.isfinite(float(‘inf’))?

a) True b) False c) None d) error

**Answer**: b

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

5. What is returned by math.isfinite(float(‘nan’))?

a) True b) False c) None d) error

**Answer**: b

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

6. What is x if x = math.isfinite(float(‘0.0’))?

a) True b) False c) None d) error

**Answer**: a

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

7. What is the result of the following?

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

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

**Answer**: b

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

8. What is the output of the following?

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

a) error, the minus sign shouldn’t have been inside the brackets

b) error, there is no function called isinf

c) True d) False

**Answer**: c

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

9. What is the value of x if x = math.ldexp(0.5, 1)?

a) 1 b) 2.0 c) 0.5 d) none of the mentioned

**Answer**: d

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

10. What is returned by math.modf(1.0)?

a) (0.0, 1.0) b) (1.0, 0.0 c) (0.5, 1) d) (0.5, 1.0)

**Answer**: a

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

**Python MCQ of** – Math – 3

This set of Python Developer Interview Questions & **Answer**s focuses on “Math – 3”.

1. What is the result of math.trunc(3.1)?

a) 3.0 b) 3 c) 0.1 d) 1

**Answer**: b

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

2. What is the output of print(math.trunc(‘3.1’))?

a) 3 b) 3.0 c) error d) none of the mentioned

**Answer**: c

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

3. Which of the following is the same as math.exp(p)?

a) e \*\* p b) math.e \*\* p

c) p \*\* e d) p \*\* math.e

**Answer**: b

***Explanation:***  math.e is the constant defined in the math module.

4. What is returned by math.expm1(p)?

a) (math.e \*\* p) – 1 b) math.e \*\* (p – 1)

c) error d) none of the mentioned

**Answer**: a

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

5. What is the default base used when math.log(x) is found?

a) e b) 10 c) 2 d) none of the mentioned

**Answer**: a

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

6. Which of the following aren’t defined in the math module?

a) log2() b) log10()

c) logx() d) none of the mentioned

**Answer**: c

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

7. What is returned by int(math.pow(3, 2))?

a) 6 b) 9

c) error, third argument required

d) error, too many arguments

**Answer**: b

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

8. What is output of print(math.pow(3, 2))?

a) 9 b) 9.0 c) None d) none of the mentioned

**Answer**: b

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

9. What is the value of x if x = math.sqrt(4)?

a) 2 b) 2.0 c) (2, -2) d) (2.0, -2.0)

**Answer**: b

***Explanation:***  The function returns one floating point number.

10. What does math.sqrt(X, Y) do?

a) calculate the Xth root of Y

b) calculate the Yth root of X

c) error

d) return a tuple with the square root of X and Y

**Answer**: c

***Explanation:***  The function takes only one argument.

**Python Question and Answers – Datetime Module – 1**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Datetime Module – 1”.**

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

import datetime

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

print(d)

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

**Answer**: b

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

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

import datetime

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

print(d)

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

**Answer**: a

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

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

tday=datetime.date.today()

print(tday.month())

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

**Answer**: d

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

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

import datetime

tday=datetime.date.today()

print(tday)

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

**Answer**: c

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

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

tday=datetime.date.today()

print(tday.weekday())

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

**Answer**: a

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

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

tday=datetime.date.today()

print(tday.isoweekday())

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

**Answer**: d

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

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

tday=datetime.date.today()

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

till\_bday=bday-tday

print(till\_bday)

a) 3 months, 0:00:00 b) 90 days, 0:00:00

c) 3 months 2 days, 0:00:00 d) 92 days, 0:00:00

**Answer**: d

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

8. The value returned when we use the function isoweekday() is \_\_\_\_\_\_ and that for the function weekday() is \_\_\_\_\_\_\_\_ if the system date is 19th June, 2017 (Monday).

a) 0,0 b) 0,1 c) 1,0 d) 1,1

**Answer**: c

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

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

tday=datetime.date.today()

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

t\_day=bday-tday

a) print(t\_day.seconds) b) print(t\_day.months)

c) print(t\_day.max) d) print(t\_day.resolution)

**Answer**: b

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

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

tday=datetime.date.today()

tdelta=datetime.timedelta(days=10)

print(tday+tdelta)

a) 2017-16-19 b) 2017-06-9

c) 2017-06-29 d) Error

**Answer**: c

***Explanation:***  The code shown above will add the specified number of days to the current date and print the new date. On adding ten days to 6/19/2017, we get 6/29/2017. Hence the output is: 2017-06-29.

**Python Question and Answers – Datetime Module – 2**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Datetime Module – 2”.**

1. The output of both of the print statements is the same. State whether true or false.

import datetime

dt\_1 = datetime.datetime.today()

dt\_2 = datetime.datetime.now()

print(dt\_1)

print(dt\_2)

a) True b) False

**Answer**: b

***Explanation:***  The output of the two print statements is not the same because of the difference in time between the execution of the two print statements. There is a difference in the order of milliseconds between the two statements and this is reflected in the output.

2. Which of the following functions can be used to find the coordinated universal time, assuming that the datetime module has already been imported?

a) datetime.utc() b) datetime.datetime.utc()

c) datetime.utcnow() d) datetime.datetime.utcnow()

**Answer**: d

***Explanation:***  The function datetime.datetime.utcnow() can be used to find the UTC (Coordinated Universal Time), assuming that the datetime module has already been imported. The other function s shown above are invalid.

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

import time

time.time()

a) The number of hours passed since 1st January, 1970

b) The number of days passed since 1st January, 1970

c) The number of seconds passed since 1st January, 1970

d) The number of minutes passed since 1st January, 1970

**Answer**: c

***Explanation:***  The code shown above will return the number of seconds passed since 1st January, 1970.

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

def num(m):

t1 = time.time()

for i in range(0,m):

print(i)

t2 = time.time()

print(str(t2-t1))

num(3)

a) 1 and The time taken for the execution of the code

b) 3 and The time taken for the execution of the code

c) 1 and UTC time

d) 3 and UTC time

**Answer**: a

***Explanation:***  The code shown above will return the numbers 1, 2, 3, followed by the time taken in the execution of the code. Hence option (a) shows the output correctly.

5. The output of the code shown below:

import time

time.asctime()

a) Current date only b) UTC time

c) Current date and time d) Current time only

**Answer**: c

***Explanation:***  The function time.asctime(), present if the time module can be used to return the current date and time. It can also accept a parameter and return the date and time in a particular format. However in the above code, since we have not passed any parameters in the above code, the current date and time is returned.

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

import time

t=(2010, 9, 20, 8, 15, 12, 6)

time.asctime(t)

a) ‘20 Sep 2010 8:15:12 Sun’ b) ‘2010 20 Sept 08:15:12 Sun’

c) ‘Sun Sept 20 8:15:12 2010’ d) Error

**Answer**: d

***Explanation:***  The code shown above results in an error because this function accepts exactly 9 arguments (including day of the year and DST), but only 7 are given. Hence an error is thrown.

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

import time

t=(2010, 9, 20, 8, 45, 12, 6, 0, 0)

time.asctime(t)

a) ‘Sep 20 2010 08:45:12 Sun’

b) ‘Sun Sep 20 08:45:12 2010’

c) ’20 Sep 08:45:12 Sun 2010’

d) ‘2010 20 Sep 08:45:12 Sun’

**Answer**: b

***Explanation:***  The code shown above returns the given date and time in a particular format. Hence the output of the code shown above will be: ‘Sun Sep 20 08:45:12 2010’.

8. The sleep function (under the time module) is used to:

a) Pause the code for the specified number of seconds

b) Return the specified number of seconds, in terms of milliseconds

c) Stop the execution of the code

d) Return the output of the code had it been executed earlier by the specified number of seconds

**Answer**: a

***Explanation:***  The sleep function (under the time module) is used to pause the code for the specified number of seconds. The number of seconds is taken as an argument by this function.

9. The output of the code shown will be:

import time

for i in range(0,5):

print(i)

time.sleep(2)

a) After an interval of 2 seconds, the numbers 1, 2, 3, 4, 5 are printed all together

b) After an interval of 2 seconds, the numbers 0, 1, 2, 3, 4 are printed all together

c) Prints the numbers 1, 2, 3, 4, 5 at an interval of 2 seconds between each number

d) Prints the numbers 0, 1, 2, 3, 4 at an interval of 2 seconds between each number

**Answer**: d

***Explanation:***  The output of the code shown above will be the numbers 0, 1, 2, 3, 4 at an interval of 2 seconds each.

10. The output of the code shown below is

time.struct\_time(tm\_year=2017, tm\_mon=6, tm\_mday=25, tm\_hour=18, tm\_min=26, tm\_sec=6, tm\_wday=6, tm\_yday=176, tm\_isdst=0)

Code:

import time

t=time.localtime()

print(t)

To extract only the year from this, we can use the function:

a) t[1] b) tm\_year c) t[0] d) t\_year

**Answer**: c

***Explanation:***  To extract the year from the code shown above, we use the command t[0]. The command t[1] will return the month number (6 in the above case). The commands tm\_year and t\_year will result in errors.

11. State whether true or false.

s = time.time()

t= time.time()

s == t

a) True b) False

**Answer**: b

***Explanation:***  The variables ‘s’ and ‘t’ will not be equal due to the slight difference in the time of their execution. Hence the output of this code will be: False.

**Python Question and Answers – Random module – 1**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Random module – 1”.**

1. To include the use of functions which are present in the random library, we must use the option:

a) import random b) random.h

c) import.random d) random.random

**Answer**: a

***Explanation:***  The command import random is used to import the random module, which enables us to use the functions which are present in the random library.

2. The output of the following snippet of code is either 1 or 2. State whether this statement is true or false.

import random

random.randint(1,2)

a) True

b) False

**Answer**: a

***Explanation:***  The function random.randint(a,b) helps us to generate an integer between ‘a’ and ‘b’, including ‘a’ and ‘b’. In this case, since there are no integers between 1 and 2 , the output will necessarily be either 1 or 2’.

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

import random

random.choice(2,3,4)

a) An integer other than 2, 3 and 4 b) Either 2, 3 or 4

c) Error d) 3 only

**Answer**: c

***Explanation:***  The code shown above displays the incorrect syntax of the function random.choice(). This functions takes its numeric parameter in the form of a list. Hence the correct syntax world be: random.choice([2,3,4]).

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

import random

random.choice([10.4, 56.99, 76])

a) Error b) Either 10.4, 56.99 or 76

c) Any number other than 10.4, 56.99 and 76 c) 56.99 only

**Answer**: b

***Explanation:***  The function random.choice(a,b,c,d) returns a random number which is selected from a, b, c and d. The output can be either a, b, c or d. Hence the output of the snippet of code shown above can be either 10.4, 56.99 or 76.

5. What is the output of the function shown below (random module has already been imported)?

random.choice('sun')

a) sun b) u c) either s, u or n d) error

**Answer**: c

***Explanation:***  The above function works with alphabets just as it does with numbers. The output of this expression will be either s, u or n.

6. What is the output of the following function, assuming that the random module has already been imported?

random.uniform(3,4)

a) Error b) Either 3 or 4 c) Any integer other than 3 and 4

d) Any decimal value between 3 and 4

**Answer**: d

***Explanation:***  This question depicts the basic difference between the functions random.randint(a, b) and random.uniform(a, b). While random.randint(a,b) generates an integer between ‘a’ and ‘b’, including ‘a’ and ‘b’, the function random.uniform(a,b) generates a decimal value between ‘a’ and ‘b’.

7. What is the output of the function shown below if the random module has already been imported?

random.randint(3.5,7)

a) Error

b) Any integer between 3.5 and 7, including 7

c) Any integer between 3.5 and 7, excluding 7

d) The integer closest to the mean of 3.5 and 7

**Answer**: a

***Explanation:***  The function random.randint() does not accept a decimal value as a parameter. Hence the function shown above will throw an error.

8. Which of the following functions helps us to randomize the items of a list?

a) seed b) randomize c) shuffle d) uniform

**Answer**: c

***Explanation:***  The function shuffle, which is included in the random module, helps us to randomize the items of a list. This function takes the list as a parameter.

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

random.seed(3)

random.randint(1,5)

random.seed(3)

random.randint(1,5)

a) 3 b) 2

c) Any integer between 1 and 5, including 1 and 5

d) Any integer between 1 and 5, excluding 1 and 5

**Answer**: b

***Explanation:***  We use the seed function when we want to use the same random number once again in our program. Hence the output of the code shown above will be 2, since 2 was generated previously following which we used the seed function.

10. What is the interval of the value generated by the function random.random(), assuming that the random module has already been imported?

a) (0,1) b) (0,1] c) [0,1] d) [0,1)

**Answer**: d

***Explanation:***  The function random.random() generates a random value in the interval [0,1), that is, including zero but excluding one.

11. Which of the following is a possible outcome of the function shown below?

random.randrange(0,91,5)

a) 10 b) 18 c) 79 d) 95

**Answer**: a

***Explanation:***  The function shown above will generate an output which is a multiple of 5 and is between 0 and 91. The only option which satisfies these criteria is 10. Hence the only possible output of this function is 10.

12. Both the functions randint and uniform accept \_\_\_\_\_\_\_\_\_\_\_\_ parameters.

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

**Answer**: c

***Explanation:***  Both of these functions, that is, randint and uniform are included in the random module and both of these functions accept 3 parameters. For example: random.uniform(self,a,b) where ‘a’ and ‘b’ specify the range and self is an imaginary parameter.

13. The randrange function returns only an integer value. State whether true or false.

a) True

b) False

**Answer**: a

***Explanation:***  The function randrange returns only an integer value. Hence this statement is true.

14. Which of the following options is the possible outcome of the function shown below?

random.randrange(1,100,10)

a) 32 b) 67 c) 91 d) 80

**Answer**: c

***Explanation:***  The output of this function can be any value which is a multiple of 10, plus 1. Hence a value like 11, 21, 31, 41…91 can be the output. Also, the value should necessarily be between 1 and 100. The only option which satisfies this criteria is 91.

15. What is the output of this function, assuming that the random library has already been included?

random.shuffle[1,2,24]

a) Randomized list containing the same numbers in any order

b) The same list, that is [1,2,24].

c) A list containing any random numbers between 1 and 24

d) Error

**Answer**: d

***Explanation:***  The function shown above will result in an error because this is the incorrect syntax for the usage of the function shuffle(). The list should be previously declared and then passed to this function to get an output.

An example of the correct syntax:

>>> l=[‘a’,’b’,’c’,’d’].

>>> random.shuffle(l)

>>> print(l)

**Python MCQ of – Random Module – 2**

**This set of Python Aptitude Test focuses on “Random module”.**

1. What the does random.seed(3) return?

a) True b) None c) 3 d) 1

**Answer**: b

***Explanation:***  The function random.seed() always returns a None.

2. Which of the following cannot be returned by random.randrange(4)?

a) 0 b) 3 c) 2.3 d) none of the mentioned

**Answer**: c

***Explanation:***  Only integers can be returned.

3. Which of the following is equivalent to random.randrange(3)?

a) range(3) b) random.choice(range(0, 3))

c) random.shuffle(range(3)) d) random.select(range(3))

**Answer**: b

***Explanation:***  It returns one number from the given range.

4. The function random.randint(4) can return only one of the following values. Which?

a) b) 3.4 c) error d) 5

**Answer**: c

***Explanation:***  Error, the function takes two arguments.

5. Which of the following is equivalent to random.randint(3, 6)?

a) random.choice([3, 6]) b) random.randrange(3, 6)

c) 3 + random.randrange(3) d) 3 + random.randrange(4)

**Answer**: d

***Explanation:***  random.randint(3, 6) can return any one of 3, 4, 5 and 6.

6. Which of the following will not be returned by random.choice(“1 ,”)?

a) 1 b) (space) c) , d) none of the mentioned

**Answer**: d

***Explanation:***  Any of the characters present in the string may be returned.

7. Which of the following will never be displayed on executing print(random.choice({0: 1, 2: 3}))?

a) 0 b) 1 c) KeyError: 1 d) none of the mentioned

**Answer**: a

***Explanation:***  It will not print 0 but dict[0] i.e. 1 may be printed.

8. What does random.shuffle(x) do when x = [1, 2, 3]?

a) error

b) do nothing, it is a placeholder for a function that is yet to be implemented

c) shuffle the elements of the list in-place

d) none of the mentioned

**Answer**: c

***Explanation:***  The elements of the list passed to it are shuffled in-place.

9. Which type of elements are accepted by random.shuffle()?

a) strings b) lists c) tuples d) integers

**Answer**: b

***Explanation:***  Strings and tuples are immutable and an integer has no len().

10. What is the range of values that random.random() can return?

a) [0.0, 1.0]. b) (0.0, 1.0]. c) (0.0, 1.0) d) [0.0, 1.0)

**Answer**: d

***Explanation:***  Any number that is greater than or equal to 0.0 and lesser than 1.0 can be returned.

**Python Question and Answers – Sys Module**

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

1. Which of the following functions can help us to find the version of python that we are currently working on?

a) sys.version b) sys.version()

c) sys.version(0) d) sys.version(1)

**Answer**: a

***Explanation:***  The function sys.version can help us to find the version of python that we are currently working on. For example, 3.5.2, 2.7.3 etc. this function also returns the current date, time, bits etc along with the version.

2. Which of the following functions is not defined under the sys module?

a) sys.platform b) sys.path

c) sys.readline d) sys.argv

**Answer**: c

***Explanation:***  The functions sys.platform, sys.path and sys.argv are defined under the sys module. The function sys.readline is not defined. However, sys.stdin.readline is defined.

3. The output of the functions len(“abc”) and sys.getsizeof(“abc”) will be the same. State whether true or false.

a) True b) False

**Answer**: b

***Explanation:***  The function len returns the length of the string passed, and hence it’s output will be 3. The function getsizeof, present under the sys module returns the size of the object passed. It’s output will be a value much larger than 3. Hence the above statement is false.

4. What is the output of the code shown below, if the code is run on Windows operating system?

import sys

if sys.platform[:2]== 'wi':

print("Hello")

a) Error b) Hello

c) No output d) Junk value

**Answer**: b

***Explanation:***  The output of the function sys.platform[:2] is equal to ‘wi’, when this code is run on windows operating system. Hence the output printed is ‘hello’.

5. What is the output of the following line of code, if the sys module has already been imported?

sys.stdout.write("hello world")

a) helloworld b) hello world10 c) hello world11 d) error

**Answer**: c

***Explanation:***  The function shown above prints the given string along with the length of the string. Hence the output of the function shown above will be hello world11.

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

import sys

sys.stdin.readline()

Sanfoundry

a) ‘Sanfoundry\n’ b) ‘Sanfoundry’

c) ‘Sanfoundry10’ d) Error

**Answer**: a

***Explanation:***  The function shown above works just like raw\_input. Hence it automatically adds a ‘\n’ character to the input string. Therefore, the output of the function shown above will be: Sanfoundry\n.

7. What is the output of this code?

import sys

eval(sys.stdin.readline())

"India"

a) India5 b) India c) ‘India\n’ d) ‘India’

**Answer**: d

***Explanation:***  The function shown above evaluates the input into a string. Hence if the input entered is enclosed in double quotes, the output will be enclosed in single quotes. Therefore, the output of this code is ‘India’.

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

import sys

eval(sys.stdin.readline())

Computer

a) Error b) ‘Computer\n’

c) Computer8 d) Computer

**Answer**: a

***Explanation:***  The code shown above will result in an error. This is because this particular function accepts only strings enclosed in single or double inverted quotes, or numbers. Since the string entered above is not enclosed in single or double inverted quotes, an error will be thrown.

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

import sys

sys.argv[0]

a) Junk value b) ‘ ‘ c) No output d) Error

**Answer**: b

***Explanation:***  The output of the function shown above will be a blank space enclosed in single quotes. Hence the output of the code shown above is ‘ ‘.

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

import sys

sys.stderr.write(“hello”)

a) ‘hello’ b) ‘hello\n’ c) hello d) hello5

**Answer**: d

***Explanation:***  The code shown above returns the string, followed by the length of the string. Hence the output of the code shown above is hello5.

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

import sys

sys.argv

a) ‘ ‘ b) [ ] c) [‘ ‘] d) Error

**Answer**: c

***Explanation:***  The output of the code shown above is a blank space inserted in single quotes, which is enclosed by square brackets. Hence the output will be [‘ ‘].

12. To obtain a list of all the functions defined under sys module, which of the following functions can be used?

a) print(sys) b) print(dir.sys)

c) print(dir[sys]) d) print(dir(sys))

**Answer**: d

***Explanation:***  The function print(dir(sys)) helps us to obtain a list of all the functions defined under the sys module. The function can be used to obtain the list of functions under any given module in Python.

13. The output of the function len(sys.argv) is \_\_\_\_\_\_\_\_\_\_\_\_

a) Error b) 1

c) 0 d) Junk value

**Answer**: b

***Explanation:***  The output of the function sys.argv is [‘ ‘]. When we execute the function len([‘ ‘]), the output is 1. Hence the output of the function len(sys.argv) is also 1.

**Python MCQ of – Operating System**

**This set of Python Questions & Answers for Exams focuses on “Operating System”.**

1. What does os.name contain?

a) the name of the operating system dependent module imported

b) the address of the module os

c) error, it should’ve been os.name()

d) none of the mentioned

**Answer**: a

***Explanation:***  It contains the name of the operating system dependent module imported such as ‘posix’, ‘java’ etc.

2. What does print(os.geteuid()) print?

a) the group id of the current process

b) the user id of the current process

c) both the group id and the user of the current process

d) none of the mentioned

**Answer**: b

***Explanation:***  os.geteuid() gives the user id while the os.getegid() gives the group id.

3. What does os.getlogin() return?

a) name of the current user logged in

b) name of the superuser

c) gets a form to login as a different user

d) all of the above

**Answer**: a

***Explanation:***  It returns the name of the user who is currently logged in and is running the script.

4. What does os.close(f) do?

a) terminate the process f

b) terminate the process f if f is not responding

c) close the file descriptor f

d) return an integer telling how close the file pointer is to the end of file

**Answer**: c

***Explanation:***  When a file descriptor is passed as an argument to os.close() it will be closed.

5. What does os.fchmod(fd, mode) do?

a) change permission bits of the file

b) change permission bits of the directory

c) change permission bits of either the file or the directory

d) none of the mentioned

**Answer**: a

***Explanation:***  The arguments to the function are a file descriptor and the new mode.

6. Which of the following functions can be used to read data from a file using a file descriptor?

a) os.reader()

b) os.read()

c) os.quick\_read()

d) os.scan()

**Answer**: b

***Explanation:***  None of the other functions exist.

7. Which of the following returns a string that represents the present working directory?

a) os.getcwd() b) os.cwd() c) os.getpwd() d) os.pwd()

**Answer**: a

***Explanation:***  The function getcwd() (get current working directory) returns a string that represents the present working directory.

8. What does os.link() do?

a) create a symbolic link b) create a hard link

c) create a soft link d) none of the mentioned

**Answer**: b

***Explanation:***  os.link(source, destination) will create a hard link from source to destination.

9. Which of the following can be used to create a directory?

a) os.mkdir() b) os.creat\_dir()

c) os.create\_dir() d) os.make\_dir()

**Answer**: a

***Explanation:***  The function mkdir() creates a directory in the path specified.

10. Which of the following can be used to create a symbolic link?

a) os.symlink() b) os.symb\_link()

c) os.symblin() d) os.ln()

**Answer**: a

***Explanation:***  It is the function that allows you to create a symbolic link.

**Python MCQ of – Turtle Module – 1**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Turtle Module – 1”.**

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

import turtle

t=turtle.Pen()

for i in range(0,4):

t.forward(100)

t.left(120)

a) square b) rectangle c) triangle d) kite

**Answer**: c

***Explanation:***  According to the code shown above, 4 lines will be drawn. Three lines will be in the shape of a triangle. The fourth line will trace the base, which is already drawn. Hence the base will be slightly thicker than the rest of the lines. However there will be no change in the shape due to this extra line. Hence the output shape will be a triangle.

2. The number of lines drawn in each case, assuming that the turtle module has been imported:

Case 1:

for i in range(0,10):

turtle.forward(100)

turtle.left(90)

Case 2:

for i in range(1,10):

turtle.forward(100)

turtle.left(90)

a) 10, 9 b) 9, 10 c) 9, 9 d) 10, 10

**Answer**: a

***Explanation:***  The number of lines drawn in the first case is 10, while that in the second case is 9.

3. The command which helps us to reset the pen (turtle):

a) turtle.reset b) turtle.penreset

c) turtle.penreset() d) turtle.reset()

**Answer**: d

***Explanation:***  The command turtle.reset() helps us to reset the pen. After the execution of this command, we get a blank page with an arrow on it. We can then perform any desired operation on this page.

4. Fill in the blank such that the code shown below results in the formation of an inverted, equilateral triangle.

import turtle

t=turtle.Pen()

for i in range(0,3):

t.forward(150)

t.right(\_\_\_\_\_)

a) -60 b) 120 c) -120 c) 60

**Answer**: b

***Explanation:***  An angle of -120 will result in the formation of an upright, equilateral triangle. An angle of 120 will result in the formation of an inverted triangle. The angles of 60 and -60 do not result in the formation of a triangle.

5. What will be the output shape of the code shown below?

import turtle

t=turtle.Pen()

for i in range(1,4):

t.forward(60)

t.left(90)

a) Rectangle b) Trapezium c) Triangle d) Square

**Answer**: d

***Explanation:***  The code shown above will result in the formation of a square, with each of side 60.

6. What is the output of the following code?

import turtle

t=turtle.Pen()

for i in range(0,4):

t.forward(100)

t.left(90)

t.penup()

t.left(90)

t.forward(200)

for i in range(0,4):

t.forward(100)

t.left(90)

a) Error b) 1 square

c) 2 squares, at a separation of100 units, joined by a straight line

d) 2 squares, at a separation of 100 units, without a line joining them

**Answer**: b

***Explanation:***  The output of the code shown above will be a single square. This is because the function t.penup() is used to lift the pen after the construction of the first square. However, the function t.pendown() has not been used to put the pen back down. Hence, the output shape of this code is one square, of side 100 units.

7. Which of the following functions does not accept any arguments?

a) position b) fillcolor c) goto d) setheading()

**Answer**: a

***Explanation:***  The functions fillcolor(), goto() and setheading() accept arguments, whereas the function position() does not accept any arguments. The function position() returns the current position of the turtle.

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

import turtle

t=turtle.Pen()

t.goto(300,9)

t.position()

a) 300.00, 9.00 b) 9, 300 c) 300, 9 d) 9.00, 300.00

**Answer**: a

***Explanation:***  The goto functions takes the arrow to the position specified by the user as arguments. The position function returns the current position of the arrow. Hence the output of the code shown above will be: 300.00, 9.00.

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

import turtle

t=turtle.Pen()

for i in range(0,5):

t.left(144)

t.forward(100)

a) Trapezium b) Parallelepiped c) Tetrahedron d) Star

**Answer**: d

***Explanation:***  It is clear from the above code that 5 lines will be drawn on the canvas, at an angle of 144 degrees. The only shape which fits this description is star. Hence the output of the code shown above is star.

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

import turtle

t=turtle.Pen()

for i in range(0,3):

t.forward(100)

t.left(120)

t.back(100)

for i in range(0,3):

t.forward(100)

t.left(120)

a) Error b) Two triangles, joined by a straight line

c) Two triangles, joined at one vertex

d) Two separate triangles, not connected by a line

**Answer**: c

***Explanation:***  The output of the code shown above is two equilateral triangles (of side 100 units), joined at the vertex.

**Python MCQ of – Turtle Module – 2**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Turtle Module – 2”.**

1. The output of the code shown below:

import turtle

t=turtle.Pen()

t.color(0,0,1)

t.begin\_fill()

t.circle(15)

t.end\_fill()

a) Error

b) A circle filled in with the colour red

c) A circle filled in with the colour blue

d) A circle filled in with the colour green

**Answer**: c

***Explanation:***  The function t.colour(0, 0, 1) is used to fill in the colour blue into any given shape. Hence the output of the code shown above will be a circle filled in with the colour blue.

2. Which of the following functions can be used to make the arrow black?

a) turtle.color(0,1,0) b) turtle.color(1,0,0)

c) turtle.color(0,0,1) d) turtle.color(0,0,0)

**Answer**: d

***Explanation:***  The function turtle.color(0,0,0) can change the colour of the arrow. The function turtle.color(0,1,0) will make the arrow green. The function turtle.color(1,0,0) will make the arrow red. The function turtle.color(0,0,1) will make the arrow blue. The function turtle.color(0,0,0) will make the arrow black.

3. What is the output of the code shown?

import turtle

t=turtle.Pen()

t.color(1,1,1)

t.begin\_fill()

for i in range(0,3):

t.forward(100)

t.right(120)

t.end\_fill()

a) Blank page

b) A triangle filled in with the colour yellow

c) A triangle which is not filled in with any colour

d) Error

**Answer**: a

***Explanation:***  The code shown above will result in a blank page. This is because the command turtle.color(1,1,1) eliminates the arrow from the page. Hence all the commands after this command are ineffective.

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

import turtle

t=turtle.Pen()

t.color(0,1,0)

t.begin\_fill()

for i in range(0,4):

t.forward(100)

t.right(90)

a) A square filled in with the colour green

b) A square outlined with the colour green

c) Blank canvas

d) Error

**Answer**: c

***Explanation:***  The output shape of the code shown above is a square, outlined with the colour green, but not filled in with any colour. This is because we have not used the command t.end\_fill() at the end.

5. In which direction is the turtle pointed by default?

a) North b) South c) East d) West

**Answer**: c

***Explanation:***  By default, the turtle is pointed towards the east direction. We can change the direction of the turtle by using certain commands. However, whenever the turtle is reset, it points towards east.

6. The command used to set only the x coordinate of the turtle at 45 units is:

a) reset(45) b) setx(45) c) xset(45) d) xreset(45)

**Answer**: b

***Explanation:***  The command setx(45) is used to set the x coordinate of the turtle. Similarly, the command sety() is used to set the y coordinate of the turtle. The function reset() takes two values as arguments, one for the x-coordinate and the other for the y-coordinate.

7. Which of the following functions returns a value in degrees, counterclockwise from the horizontal right?

a) heading() b) degrees()

c) position() d) window\_height()

**Answer**: a

***Explanation:***  The function heading() returns the heading of the turtle, which is a value in degrees counterclockwise from the horizontal right. This measure will be in radians if radians() has been called.

8. What is the output of the following code?

import turtle

t=turtle.Pen()

t.right(90)

t.forward(100)

t.heading()

a) 0.0 b) 90.0 c) 270.0 d) 360.0

**Answer**: c

***Explanation:***  The output of the code shown above will be 270.0. The function heading() returns the heading of the turtle, a value in degrees, counterclockwise from the horizontal right. The output shape of this code is a straight line pointing downwards.

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

import turtle

t=turtle.Pen()

t.clear()

t.isvisible()

a) Yes b) True c) No d) False

**Answer**: b

***Explanation:***  The function t.clear() returns a blank canvas, without changing the position of the turtle. Since the turtle is visible on the blank canvas, the output of this code is: Yes.

10. What is the output of the code shown?

import turtle

t=turtle.Pen()

t.forward(100)

t.left(90)

t.clear()

t.position()

a) 0.00, 90.00 b) 0.00, 0.00

c) 100.00, 90.00 d) 100.00, 100.00

**Answer**: d

***Explanation:***  The output of the code shown above is 100.00, 100.00. The function clear() is used to erase the entire canvas and redraw the turtle. However, the position of the turtle is not changed.

**Python MCQ of – Turtle Module – 3**

**This set of Python Multiple Choice Questions & Answers (MCQs)**

**focuses on “Turtle Module – 3”.**

1. Which of the following functions results in an error?

a) turtle.shape(“turtle”) b) turtle.shape(“square”)

c) turtle.shape(“triangle”) d) turtle.shape(“rectangle”)

**Answer**: d

***Explanation:***  The functions shown above will change the arrow to the shape mentioned. The functions turtle.shape(“turtle”), turtle.shape(“square”) and turtle.shape(“triangle”) are valid whereas the function turtle.shape(“rectangle”) is invalid.

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

import turtle

t=turtle.Pen

t.tilt(75)

t.forward(100)

a) A straight line of 100 units tiled at 75 degrees from the horizontal

b) A straight line of 100 units tilted at 15 degrees from the horizontal

c) A straight line of 100 units lying along the horizontal

d) Error

**Answer**: c

***Explanation:***  The function turtle.tilt(75) will tilt the turtle. But the straight line (of 100 units) is drawn along the horizontal. Hence the output of the code shown above is a straight line of 100 units lying along the horizontal.

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

import turtle

t=turtle.Pen()

t.backward(100)

t.penup()

t.right(45)

t.isdown()

a) True b) False c) Yes d) No

**Answer**: b

***Explanation:***  In the code shown above, we have used the function t.penup() to life the pen from the canvas. However, we have not used the function t.pendown() to keep the pen back down. The function turtle.isdown() returns True if the pen is down and False if the pen is not down. Hence the output is False.

4. The function used to alter the thickness of the pen to ‘x’ units:

a) turtle.width(x) b) turtle.span(x)

c) turtle.girth(x) d) turtle.thickness(x)

**Answer**: a

***Explanation:***  The function turtle.width(x) is used to alter the thickness of the pen to ‘x’ units. The function turtle.span(x), turtle.girth(x) and turtle.thickness(x) are invalid.

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

import turtle

t=turtle.Pen()

t.goto(100,0)

t.towards(0,0)

a) 0.0 b) 180.0 c) 270.0 d) 360.0

**Answer**: b

***Explanation:***  The function t.towards(x,y) returns the angle between the line to the line specified by (x,y). Hence the output will be 180.0.

6. What is the output of the following code?

import turtle

t=turtle.Pen()

t.position()

(100.00,0.00)

t.goto(100,100)

t.distance(100,0)

a) 0.0 b) Error c) 100.0, 100.0 d) 100.0

**Answer**: d

***Explanation:***  The distance() function returns the distance between the turtle to the given vector. Hence the output of the code shown above is 100.0.

7. The output of the following code will result in a shape similar to the alphabet \_\_\_\_\_\_\_\_\_\_\_

import turtle

t=turtle.Turtle()

t1=turtle.Turtle()

t.left(45)

t1.left(135)

t.forward(100)

t1.forward(100)

a) V b) Inverted V c) X d) T

**Answer**: a

***Explanation:***  In the code shown above, two pens have been used to create a shape similar to the alphabet ‘V’. The angle between the two straight lines is 90 degrees.

8. The output of the code shown is similar to the alphabet \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import turtle

t=turtle.Pen()

t1=turtle.Pen()

t2=turtle.Pen()

t.forward(100)

t1.forward(100)

t2.forward(100)

t1.left(90)

t1.forward(75)

t2.right(90)

t2.forward(75)

a) X b) N c) T d) M

**Answer**: c

***Explanation:***  In the above code, three pens have been used to create a shape similar to the letter ‘T’. All the three straight lines are mutually perpendicular.

9. The code shown below will result in an error. State whether true or false.

import turtle

t=turtle.Pen()

t.speed(-45)

t.circle(30)

a) True b) False

**Answer**: b

***Explanation:***  Although a negative speed is not possible, the code shown above does not result in an error. Hence, the **Answer** is False.

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

import turtle()

t=turtle.Pen()

t.goto(50,60)

t1=t.clone()

t1.ycor()

a) 0.0 b) 50.0 c) 60.0 d) Error

**Answer**: c

***Explanation:***  The function clone() is used to create a clone of the turtle, having the same properties such as position, coordinates etc. Hence, the properties of the t and t1 are the same in the code shown above. The function ycor() returns the y-coordinate of the turtle. Hence the output of the code is 60.0.

11. The output shape of the code shown above:

import turtle

t=turtle.Pen()

for i in range(0,6):

t.forward(100)

t.left(60)

a) Hexagon b) Octagon c) Pentagon d) Heptagon

**Answer**: a

***Explanation:***  The code shown above creates a six-sided polygon. The output shape of the code shown above is will be a hexagon.

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

import turtle

t=turtle.Pen()

t.resizemode(“sanfoundry”)

t.resizemode()

a) user b) auto c) nonresize d) error

**Answer**: c

***Explanation:***  When not explicitly specified as auto or user, no adaption of the turtle’s appearance takes place and the mode is ‘noresize’. Hence the output of the code is: noresize.

**Python MCQ of – 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”)

**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”)

**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”)

**Answer**: a

***Explanation:***  a is used to indicate that data is to be appended.

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

**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()

**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()

**Answer**: b

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

7. What is the output?

f = None

for i in range (5):

with open("data.txt", "w") as f:

if i > 2:

break

print(f.closed)

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

**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()

**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()

**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

**Answer**: b

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

**Python MCQ of – 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

**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?

str = raw\_input("Enter your input: ");

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

**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?

str = input("Enter your input: ");

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

**Answer**: a

***Explanation:***  None.

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

a) closed b) softspace c) rename d) mode

**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

**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

**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

**Answer**: a

***Explanation:***  remove(file\_name)

8. What is the output of this program?

fo = open("foo.txt", "rw+")

print "Name of the file: ", fo.name

# Assuming file has following 5 lines

# This is 1st line

# This is 2nd line

# This is 3rd line

# This is 4th line

# This is 5th line

for index in range(5):

line = fo.next()

print "Line No %d - %s" % (index, line)

# Close opened file

fo.close()

a) Compilation Error b) Syntax Error

c) Displays Output d) None of the mentioned

**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

**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 size

b) deletes the content of the file

c) deletes the file size

d) none of the mentioned

**Answer**: a

***Explanation:***  The method truncate() truncates the file 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 MCQ of – 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

**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?

import sys

print 'Enter your name: ',

name = ''

while True:

c = sys.stdin.read(1)

if c == '\n':

break

name = name + c

print 'Your name is:', name

If entered name is

sanfoundry

a) sanfoundry b) sanfoundry, sanfoundry

c) San d) None of the mentioned

**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 prompt 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?

import sys

sys.stdout.write(' Hello\n')

sys.stdout.write('Python\n')

a) Compilation Error b) Runtime Error

c) Hello Python d) Hello

Python

**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

**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 deserialization

c) None of the mentioned

d) All of the mentioned

**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

**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

**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?

fo = open("foo.txt", "wb")

print "Name of the file: ", fo.name

fo.flush()

fo.close()

a) Compilation Error b) Runtime Error

c) No Output d) Flushes the file when closing them

**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

**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

**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 MCQ of – 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

**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

**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

**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()

**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()

**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()

**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()

**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()

**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

**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+

**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 MCQ of – 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+

**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

**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\_\_()

**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

**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\_\_()

**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

**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)

**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’)

**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

**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

**Answer**: d

***Explanation:***  seek() takes at least one argument.

**Python MCQ of – 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

**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

**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

**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

**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

**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

**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

**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

**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 occurred")

except "someError":

print ("someError has occurred")

a) someError has occurred

b) someError has not occurred

c) invalid code

d) none of the mentioned

**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

**Answer**: b

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

**Python MCQ of – 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

**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

**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

**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 and 10 and 12 d) No output

**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 and test and 12

c) 11 and test d) 11

**Answer**: b

***Explanation:***  The code shown above results in the output:

11 and test and 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

**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]

**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

**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

**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’>

**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 MCQ of – 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

**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

**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

**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

**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

**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

**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”)

**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)

**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

**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

**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

**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

**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

**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

**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

**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.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*