Python Question and Answers – 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

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

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

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 maximumd) 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

9.
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

.

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

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

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

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

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

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 – 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 Module – 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 Module – 2

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

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 – 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 76c) 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

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

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 world10c) hello world11d) 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

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 – Files – 1

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

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) renamed) 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

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

Python MCQ of – Files – 4

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) Readb) Writec) Appendd) 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

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

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.