Who all are the user of the Python Programming?

Ans: Python programming is not only for the IT developers, it is mainly used by professionals who are working in Finance domain, Analytics, Data Scientists, Researchers and IT Developers etc.

Q1. Is Python High level or low level language?

Ans: Python is a high level language as other programming language C, C++, Java, Scala etc.

Q2. What do you mean by low level language?

Ans: Low level language which can be understood by underline machine, you can say machine language or assembly language are low level language. And computers can run the program which is written in low language only.

Q3. How high level language are executed by computers?

Ans: High level language are first needs to be converted to low level language e.g. Java Virtual Machine does of java language. This is an extra step which makes executing code slower compare to low level language.

Q4. What are the benefits of high level language?

Ans : Writing program using high level language can have following advantages.

- Easier to program
- They are portable, e.g. write program on Windows OS that can be executed on IOS without any or very less change. In case of low level language, you have to write program again for different OS.

Q5. How high level language are converted to low level language?

Ans: There are two way by which high level languages are converted into low level language.

- Interpreter
- Compiler

Q6. What is interpreter?

Ans: It does not do lot of conversion, it reads the program and directly execute on the computer. Hence, this is relatively faster than compiler.

Q7. What is compiler?

Ans: Compiler reads your program written in high level language e.g. Java Code and it is known as source file and once source file is compiled it will be converted into object code(also executable). This executable need not be converted in any other form. Now you will have run time environment where you can run this executable on any operating system.

Q8. Is Python an Interpreter or compiler language?

Ans: Python is an interpreter language. Hence, it is not required to be compiled and then executer (as in case of Java you have to do this).

Q9. What is the

extension of the Python source file?

Ans : Python source file has .py extension like test.py , is a python file.

Q10. What all types of errors in a Python program?

Ans: There are mainly three types of errors.

- Syntax Error: It is any syntactical error in the program. E.g. (8.value, is not correct)
- Runtime Error
- Semantic Error

Q11. What is a runtime error?

Ans: These errors only appears when you run the program. This is also known as exceptions.

Q12. What is semantic errors?

Ans: Catching such errors are not easy. Until and unless you test the output. It means your program will run without any issue. It will successfully complete. But when you see the result than only you can say that there is an issue e.g. You want to add two numbers 2 and 3 and expected answer is 5, but program generate 6 (because instead of + sign, * has been used.).

Q13. What is the difference between interpreter and compiler?

Ans: Interpreter convert high level language line by line but for compiler entire program will be translated in low level language.

Q14. What do you mean by data type in Python?

Ans: This is a category of the data e.g. 23 is an int , 23.7 id a float and Amit is a string type.

Q15. What is the variable in a python?

Ans: Its name to a value. You can refer a particular value using variable name like Amit it is a value but will be referred using a variable called name

Q16. What is a function?

Ans: Function is a collection of statements, which is defined once and then call it later to do some executions. Below is the function definition example

```
def total_course_durations(c1,c2):
    training=curse_duration()
    training.hours=c1.hours+c2.hours
    training.minutes=c1.minutes+c2.minutes
    training.seconds=c1.seconds+c2.seconds
    return training
```

```
total_duration=total_course_durations(hadoop_training,spark_training)
```

Name of the function: total_course_duration

Statements: All orange color lines are statements.
return: It is a return statements.

total_duration: It is a variable, holding the values return by function call.

Function Arguments: These two variables are known as function arguments
hadoop_training,spark_training

Q17. How do you convert data from one data type to another data types?

Ans: There are various functions available, which can help you convert data from one data types to another data type. See the example below

int(100) -> 100 #Converting string to int

flaot(100.5) -> 100.5 #Converting string to float

str(3.02) -> ‘3.02’ #Converting float to string

Q18. What is the module in Python?

Ans: Python module is any python file with .py extension, which contains functions and variables are known as Python Module. To use module in our current program we have to import it first. There are in-built modules which are provided by Python itself e.g. import math (Here, math is a Python module)

Q19. What do you mean by a fruitful functions?

Ans: Function, which return a value is known as fruitful function. E.g math.sqrt(4), it will return 2

Q20. What is void function?

Ans: Function which does not return a value is known as void function.

Q21. What are the all ways, by which you can import a module in Python?

Ans: Python provides two ways by which you can import the Python module.

- Using import statement e.g. import math, it will give you an object of math module.
- Another way, import only, which you want from module e.g**. <u>from math import pi</u>** now you can use pi directly without dot notation.

Q22. How can I import the all the functions and variable from a module?

Ans: You can import everything from your module using * operator as below.

from math import *

However, this is a bad practice. It will overwrite all the variables and functions which you have defined in your program. So it is suggested, to avoid this.

Q23. What is function definition and function object?

Ans : Function definition, is the way by which you define a function as below.

```
def total_course_durations(c1,c2):
  training=curse_duration()
  training.hours=c1.hours+c2.hours
  training.minutes=c1.minutes+c2.minutes
  training.seconds=c1.seconds+c2.seconds
  return training
```

Function Object: This is created by a function definition. Name of the function **total_course_durations** is a variable, which points to function object.

Q24. What do you mean by function arguments and parameters?

Ans: Parameters are defined by the names that appear in a function definition, whereas arguments are the values actually passed to a function when calling it. Parameters define what types of arguments a function can accept. For example, given the function definition:

```
def func1(value1, value2=None, **kwargs):
```

Value1, value2 and kwargs are parameters of func1. However, when calling func1, for example:

```
func(100, value2=hadoopexam, new_name=anyvar)
```

the values 100, hadoopexam and anyvar are arguments.

Q25. What do you mean by local variables?

Ans: A variable which you define inside the function is known as local variable, you can not use local variable outside the function. Below two variables course1 and course2 are local and cannot be accessed outside the function.

```
def total_course_durations(c1,c2):
    course1=c1
    course2=c2
```

Q26. What do you mean by module object?

Ans: A value created by an import statement that provides access to the values defined in a module. e.g import math , here math is a module object and using dot notation you can access the variables defined in this module for instance math.pi

Q27. What is the encapsulation?
Ans: It is the process of transforming a sequence of statements into a function definition. Everything is encapsulated inside the function.

Q28. What is the docstring in python?

Ans: Python documentation strings (or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods.

An object's docstring is defined by including a string constant as the first statement in the object's definition.

```
def my_function():
   """Do nothing, but document it.
   No, really, it doesn't do anything.
   """
```

You can print this doc string using: print my_function.__doc__

Q29. Which operator will help you get the remainder and quotient value in Python?

Ans: There are two mathematical operator to help you get quotient and remainder.

Getting remainder using % modulus operator and / to get the quotient value for example

```
quotient = 10 / 3 # will result 3
remainder = 10 % 3 #will result 1
```

Q30. What all are the relational operators?

Ans: Below are all considered relational operator to do the comparison.

```
A !=B # A is not equal to B
A>B # A is greater than B
A<B # A is less than B
A>=B # A is greater and equal to B
A<=B #A is less than or equal to B
```

Q31. What all are logical operators in Python?

Ans: There are three logical operators in Python **and, or** and **not.** See example below

- a>0 and a<10 # This condition is true only if a is greater than 0 and less than 10 . If a=100 than it will return false

- a>0 or a<10 # this condition will return true, if either a is greater than 0 or less than 10 e.g. a=100 than also it will return true.

- not (10 >100) # It will return true. Because first 10 >100, it means it is false but we are negating this condition. Hence, it will become true.

Q32. What is recursion?

Ans: A function calling itself. If you don't have a base condition (to break the recursion). It can be infinite call which can lead your program to run indefinitely and then finally crash.

Q33. What do you mean by dead code?

Ans: A part of the code, which is never executed is known as dead code. This is generally written after the return statement and will never be executed.

Q34. What is the length of empty string?
Ans: Length of empty string will always be 0 and it will be represented as.

Q35. What does it mean string is an immutable object?
Ans : String is considered an immutable object because. Once, you create it you cannot change its value.

A=Amit
a.replace(A,$) # this will not work and give error.

Q36. What is the method?
Ans: Method is a function only, but it is associated with the object. And can be called using dot notation. object.method()

Q37. How do you represent file in Python?
Ans: A file is represented using file object. For instance you have a file c:/hadoopexam.txt . If you want to read or write in this file you have to use Python file object.

Q38. What is the list in Python?
Ans: In python list is a sequence of values e.g. [Amit,Rakesh,Rohit,Umesh] is a list and contains 4 elements. List is represented using [] bracket. And its indexing start with 0. So here value at index 0 is Amit and value at index 3 is Umesh and there is no value at index 4.

Q39. Can a list contain another list?
Ans: Yes, a list can contain another list as below

[Amit,Rakesh,Rohit,Umesh , [Mumbai , Delhi , Banglore ,Chennai]]

Q40. What is an accumulator?
Ans : Accumulator is a variable, which generally used in the loop to accumulate the values in each iteration. E.g. if you want to add all the values in a list [1,2,3,4] . Then, you will be creating an accumulator which will hold the sum till previous iteration.

Q41. What is the dictionary in Python?
Ans: Dictionary is a key value pair and will be used inside this {}. For instance, see below name as key and age as value.

{Amit:32, Rakesh:37, Jayesh:42}

You cannot have duplicate keys in dictionary. Also, key should be immutable.

Q42. What is global variable?
Ans: A variable, which can be defined outside the function and can be accessed from any function is known as global variables.

Q43. Which module, you can use to debug Python code?

Ans : There is a module called pdb, which you can use for debugging.

Q44. Which tool you can use, which can help you to find the bugs in your Python source code?
Ans: You can use PyChecker tool, this will help you find the issue with your source code and also help you to do static analysis for Python.

Q45. Is there any tool or library which I can use to convert my Python code as windows executables or windows exe?
Ans: You can use py2exe tool, it convert the Python scripts into windows executables or exe and to run it you don't have to explicitly install the python on your windows machine.

Q46. What is UnboundLocalError and why you get it?
Ans: UnboundLocalError is a error when , you try to assign a local variable which is not initialized.

Let's check following steps

```
name='Amit'
def fullname():
   print name

fullname() # call the function, it will work fine.
```

Now define the function as below

```
name='Amit'
def fullname():
   print name
   name= name+'Khanna'

fullname() # call the function, it will give error "UnboundLocalError"
```

Reason: This is because when you make an assignment to a variable in a scope, that variable becomes local to that scope and shadows any similarly named variable in the outer scope. Since the last statement in fullname assigns a new value to name, the compiler recognizes it as a local variable. Consequently when the earlier print name attempts to print the uninitialized local variable and an error results.

Q47. How can be avoided "UnboundLocalError" with the global variable?
Ans: You have to mark the variable global explicitly

```
def  fullname():
   global  name
   print name
   name= name+'Khanna'
   print name

fullname()
```

Q48. I am still confused with the local and global variables of Python?
Ans : If you have defined variable outside the function and only referencing them inside the function is considered global variable. But if you try to assign them with same or new value then it will be considered a local variable and local variable must be initialized first to use it else you will get **UnboundLocalError**

Q49. You are having below code with the Lambda

```
doubled= []
for v1 in range(5) :
        doubled.append(lambda:2*v1)
print(doubled[0](), doubled[1](), doubled[2](), doubled[3]() , doubled[4]())
```

Why all the elements in doubled list are 10

Ans: Because v1 is not a local variable and it is defined in the outer scope and will be accessed in lambda function. Hence, when the loop ends, value of the v1 variable will be 4. Hence, all the values in the list will become 8.

To avoid this, you have to define a new variable that is local to the lambda as below

```
doubled= []
for v1 in range(5) :
        doubled.append(lambda x=v1:2*x)
print(doubled[0](), doubled[1](), doubled[2](), doubled[3]() , doubled[4]())
```

Now, here x is local variable to lambda, which holds the current value and then doubled and then appended to list.

Q50. How to make a variable shared across all the modules in your application and it should be global?
Ans: If you want to make a variable global across all the modules in your application than you have to define a new module usually config.py and then define that global variable in that module.

Import the config.py module in each of your application module. And as you know, only one instance of a particular module will be created. Hence, any change you make to global variable in any module will be reflected in each module. See below example with three different module files

config.py

```
price=1000 # Default price for the course will be 1000INR
```

hadoop.py

```
import config
config.price=2000 #Increase the price to 2000INR and same will be reflected in each module
```

course_detail.py

```
import config
import hadoop
print(config.price)
```

Q51. Why should you avoid below import statement
*from nameofmodule import **

Ans : Because, if you have created a variable with the same name as defined in import module as well, then it will overwrite your variable and it will be difficult to debug, what happened to the variable.

Q52. What is the best order to import the modules?
Ans : It is considered best practice, if you import the modules in following order

- First import standard library module e.g. sys, os, re
- Then third-party modules e.g. mx.DateTime
- Then modules developed by you locally.

Q53. What is the circular import and which form of circular import will not create a problem?
Ans : When you have import statement at the top and uses the import nameofmodule it is fine.

a.py

```
import b
```

b.py

```
import a
```

This is circular import and they will work fine, without any issue.

Q54. Which form of circular import will not work?
Ans : When you put circular import in the form of from nameofmodule import something and you will have circular dependencies than it will fail.

a.py

```
from b import x
```

b.py

```
from a import y
```

Here, module a is busy importing module b and in module b value y from module a is not yet available. Hence, this kind of circular import should be avoided at the top level and needs to move import to function or class level.

Only move imports into a local scope, such as inside a function definition, if it's necessary to solve a problem such as avoiding a circular import or are trying to reduce the initialization time of a module. This technique is especially helpful if many of the imports are unnecessary depending on how the program executes. You may also want to move imports into a function if the modules are only ever used in that function. Note that loading a module the first time may be expensive because of the one time initialization of the module, but loading a module multiple times is virtually free, costing only a couple of dictionary lookups. Even if the module name has gone out of scope, the module is probably available in sys.modules.

Q55. When you define a function with default shared value like def function_name(mydict={}): what issue you will see?

Ans: When you define a function as below

```
def fun1(my_dict={}):
        ... do soŵethiŶg...
        my_dict[key1]=val1
        return my_dict
```

Here, it is expected that every time you call the function fun1, it will create a new my_dict instance with no values. But this is not correct, because default values are created only once. And any subsequent call to that function will use the already created my_dict object.

To avoid such scenario, you should have used None as a default value and inside the function you should check whether the value is None then create a new my_dict.

```
def fun1(my_dict=None):
        if my_dict is None:
                my_dict={} #now it is a local dictionary object
```

Q56. How, can you pass optional or keyword parameters from one function to another function?
Ans: You have to collect the arguments as * and ** , which will give you positional arguments as a tuple and the keyword arguments as a dictionary. You can pass these arguments when calling another function using the * and **

```
def fun1(param1, *args, *kwargs):
      kwargs[key]=value
      fun2(param1, *args, *kwargs) # Calling another function
```

Q57. You have the following code

```
list1 = [ ]
list2 = list1

list2.append(ħadoopexam.com)
```

What all the values are in list1 and list2?

Ans : Both list1 and list2 will hold the same value. Because, list2 is pointing to the same list as created by list1. Hence, any modification you do either through list1 or list2 , changes will be reflected in both. In fact there is only one list and pointed by two variables named list1 and list2.

Q58. You have been given below code

```
X=100
y=x
x=x+1 #there new object of x will be created as x itself cannot be mutated.
```

What is the value hold by variable x and y?

Ans : Here x will have 101 and y will be 100. Because, integers are immutable and when you do x=x+1 , it will be creating new int object and variable x will be assigned that variable. However, y will still hold the old value.

Q59. You have a list y as below

```
y=["hadoopexam" , "learning" ,"resource" , "spark" , "learning" , "resources"]
x=y.sort()
```

When you call y.sort() , what will happen?

Ans: In this case y will hold the sorted list , but variable x is pointing to None. Most of the cases the operation which mutate object itself return None. Here, y itself sorted.

Q60. You have below code

```
x=(1,2,3)
x+=(100,)
```

What will happen to the x?

Ans : In this case x is immutable, hence new object of x will be created.

Q61. How do you find, if two variables are pointing to the same object or not?
Ans: You should have used is operator or build-in function id().

Q62. How can you return more than 1 value from a function?
Ans : You should have used tuple as a return value. If you want more than 1 value as a return value.

Q63. What do you mean by Higher-order function?
Ans : A function, which can accept other function as an arguments and return function to the caller is known as higher order fi=unction. See the example below.

```
def doOperation(a,b):
   def operation(x):
      return (a+x) *(b+x)
   return operation
```

Here, doOperation function take two arguments a and b as input and return operation function as an output. So you call the function as below.

```
result=doOperation(5,10)
result(10)
```

It will give output as below
(5+10)*(10+10)=(15)*(20)=300

Q64. What is the ideal way to copy sequences?
Ans : You should use slicing to copy sequence as below

New_list=old_list[:]

Q65. Which method will you use to find all the methods and attributes of an Object of a class?
Ans : We will be using either help() or dir() method.

Q66. How is the below statement will be evaluated?
x in a , x
Ans : Please note that comma is not an operator. It is a separator. Hence, above statement will be evaluated as below.

(x in a) , x

And result will be as below

(False, x)

Q67. What is the value of below expression?

X=100
Y=20

Val = X if X < Y else Y

Ans: Here Val will have 20. Because first condition is false, which will result in Y, which is 20.

Q68. How will you remove the duplicate elements from the list?
Ans: It's very simple, as we know set contain only unique values. Hence, first convert list to set and then set back to list.

```
list(set(list_with_duplicates))
```

Q69. What is a Class?
Ans : You can say, it is a user defined object. It works as a template and hold attributes and functions.

Q70. What is self?
Ans: Self is a first argument of a method. A method defined as do_something(self, a, b, c) should be called as object.do_something(a, b, c) for some instance object of the class in which the definition occurs; the called method will think it is called as do_something(self, a, b, c).

Q71. How would you find the current module name?
Ans : You can use __name__ to find the name of current module. But if it is run by script than , __name__ will be equal to __main__

Q72. What is the difference between dir and help function?
Ans : Both the function will help you get the detail about modules. There usage are below.

- If you write only dir, without any () braces , it will return all the names in current scope.

Using dir() function for each of the below behaves like

- For module: It will return all the module's attributes and functions.
- For Class object: It will return all the attributes of the same class as well as the base class attributes.

- dir() Returns the attributes of the object or module.
- help() Returns the python built in documentation about the object.
- type() Returns the type of object.
- __doc__ Returns the doc-string of object or module.

help() : Help function, name itself defines the usage of this function. This function returns the help related to python module, object or method if it is called with respective argument but without any argument it will return the help related to currently running programming module.

Q73. Can we override the dir() function behavior?
Ans: Yes, you can. By overriding __dir__ function.

Q74. How would you get builtin function using dir()
Ans: You can get all the builtin functions using dir(__builtins__)

Q75. Why Python does not de-allocate all the memory on exits?
Ans: Python does not de-allocate all the memory as soon as it exits, because some memory is reserved by C library and possible there could be some circular references exists. Hence, memory can not be de-allocated immediately in Python.

Q76. What zip() function does?
Ans : zip() function will combine multiple lists elements based on their position and create tuple out of this. See below example

```
l1=[1,2,3]
l2=['a','b','c','d']
l3=['hadoop' , 'exam' , 'learning']

zip(l1,l2,l3)
Out[11]:
[(1, 'a', 'hadoop'), (2, 'b', 'exam'), (3, 'c', 'learning')]
```

It will create tuple only the minimum length of the any list.

Q77. What is the Pass by Value and Pass by Reference?

Ans : Whenever arguments passed to the function they are always passed by reference. It means, if you change the value of the function inside the parameter, it will also change the value of the original parameter. However, if you are passing immutable variable then it will be pass by value, hence original copy of the variable will not be changed.

Q78. Is everything object in Python?

Ans : Everything in Python is an object, and almost everything has attributes and methods. All functions have a built-in attribute __doc__, which returns the doc string defined in the function's source code. The sys module is an object which has (among other things) an attribute called path.

Q79. What is an id() function?

Ans : id() unction provide the unique integer value of an object.

Q80. What is the purpose of the __init__ method?

Ans : The __init__ method is run as soon as an object of a class is instantiated. The method is useful to do any *initialization* you want to do with your object.

Q81. What is the purpose of pass in Python?

Ans: Suppose you are designing a new class with some methods that you don't want to implement, yet.

```
class MyClass(object):
   def meth_a(self):
      pass
   def meth_b(self):
      print "I'm meth_b"
```

If you would leave out the pass, the code wouldn't run. To summarize, the pass statement does nothing particular but can act as a placeholder.
However, there are many more usage but in short that is.

Q82. Which all are Python web scrapping libraries?

Ans : There are following Python web scrapping libraries are available

- The Farm: Requests
- The Stew: Beautiful Soup 4
- The Salad: lxml
- The Restaurant: Selenium
- The Chef: Scrapy

Q83. What is the purpose of the with statement in Python?

Ans : In python the with keyword is used when working with unmanaged resources (like file streams). It allows you to ensure that a resource is "cleaned up" when the code that uses it finishes running, even if exceptions are thrown. It provides 'syntactic sugar' for try/finally blocks.

It̕s handy when you have two related operations which you̕d like to execute as a pair, with a block of code in between. The classic example is opening a file, manipulating the file, then closing it:

```
 with open('HadoopExam.txt', 'w') as f:
    f.write('Welcoŵe to HadoopExaŵ LearŶiŶg Resources…')
```

The above with statement will automatically close the file after the nested block of code. (Continue reading to see exactly how the close occurs.) The advantage of using a with statement is that it is guaranteed to close the file no matter how the nested block exits. If an exception occurs before the end of the block, it will close the file before the exception is caught by an outer exception handler. If the nested block were to contain a return statement, or a continue or break statement, the with statement would automatically close the file in those cases, too.

Q84. What is the difference between {} dictionary and OrderedDict?

Ans: OrderedDict maintain the insertion order. In whatever order all the key value pairs are added to dictionary will be maintained and that cannot be possible with the regular dictionary.

Q85. Which is faster when we need to apply search operation on list and dictionary?

Ans : Dictionary will be faster than list, if you search based on the key.

Q86. What is the purpose of the enumerate in Python?

Ans : The enumerate() function adds a counter to an iterable. So for each element in cursor, a tuple is produced with (counter, element); the for loop binds that to row_number and row, respectively.

```
elements = ('foo', 'bar', 'baz')
for count, elem in enumerate(elements):
...     print count, elem
...
0 foo
1 bar
2 baz
```

Q87. What is a regular expression?

Ans : A regular expression (regex or regexp for short) is a special text string for describing a search pattern. You can think of regular expressions as wildcards on steroids. You are probably familiar with wildcard notations such as *.txt to find all text files in a file manager.

Q88. What is NumPy?

Ans: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors.

Q89. What is the matplotlib?

Ans: matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. SciPy makes use of matplotlib.

Q90. What is TkInter?
Ans: Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python. The name Tkinter comes from Tk interface.

Q91. What is the purpose of magic method __init__?

Ans : you don't have to invoke it directly. The invocation is realized behind the scenes. When you create an instance x of a class A with the statement "x = A()", Python will do the necessary calls to __new__ and __init__.

Q92. If Python fully object oriented language?
Ans : No, Python doesn't support strong encapsulation, which is only one of many features associated with the term "object-oriented".

The answer is simply philosophy. Guido doesn't like hiding things, and many in the Python community agree with him.

Q93. How can you ignore an exception?
Ans : Using pass , see below example

```
try:
doSomething()
except:
  pass
```

All Products List of www.HadoopExam.com

TRAINING'S (AVAILABLE)

- Hadoop BigData Professional Training
- HBase (NoSQL) Professional Training
- Apache Spark Professional Training
- Apache OOZie (Hadoop workflow) Professional Training
- Beginner AWS Training Course- **(HETRNAWS101**)
- Core Java 1z0-808 Exam training
- JAX-WS (Java WebService HandsOn Training)
- Scala Programming Training
- Python Programming Training
- Hortonworks Administration Professional Trainings

MAPR HADOOP AND NOSQL CERTIFICATION (AVAILABLE)

- **MapR Hadoop Developer Certification**
- **MapR HBase NoSQL Certification**

- **MapR Spark Developer Certification (In Progress)**

HORTONWORKS HADOOP AND NOSQL CERTIFICATION (AVAILABLE)

- HDPCD : NO Java (Hortonworks Developer Certification)
- HDPCD : Spark (Spark Developer Certifications)
- HDPCA : Hortonworks Administration Certification
- Hortonworks Administration Professional Trainings

CLOUDERA HADOOP AND SPARK CERTIFICATION (AVAILABLE)

- CCA131 : Hadoop Administrator
- CCA-175 Cloudera® (Hadoop and Spark Developer)
- CCP:DE575 : Cloudera® Data Engineer Certification
- CCA159 : Cloudera Data Analyst Certifications

DATABRICKSA OREILLY SPARK CERTIFICATION (AVAILABLE)

- Apache Spark Developer

AWS: AMAZON WEBSERVICE CERTIFICATION (AVAILABLE)

- AWS Solution Architect : Associate
- AWS Solution Architect: Professional
- AWS Developer : Associate
- AWS Sysops Admin : Associate

MICROSOFT AZURE CERTIFICATION (AVAILABLE)

- Azure 70-532
- Azure 70-533

DATA SCIENCE CERTIFICATION (AVAILABLE)

- EMC E20-007

EMC CERTIFICATIONS (AVAILABLE)

- EMC E20-007

SAS ANALYTICS CERTIFICATION (AVAILABLE)

- SAS Base A00-211
- SAS Advanced A00-212
- SAS Analytics : A00-240
- SAS Administrator : A00-250