**Python**

1. What is Python and why is it popular?

Python:

It is an interpretor high-level server side programming language and is used to create web development.

Why popular:

It is easy to learn because of easy understanding syntax.

Code maintenance is easy.

Can be used in a wide variety of technology like Machine Learning, Automation Testing, Data analysis, etc..

1. What are the differences between Python 2 and Python 3?

| Features | Python 2 | Python 3 |
| --- | --- | --- |
| Performance | Low performance due to design flaws | Better performance than python 2 |
| Syntax | More complex and difficult to interpret | Readable and easily understandable |
| Syntax - Print | It will take print as statement  Eg: print “Hello” | It will take print as function  Eg: print (“Hello”) |
| Integer Division | It will provide the result as whole number by round of the value Eg: 2 / 5 = 2 | It will provide the result as exact value Eg: 2 / 5 = 2.5 Also we get result as round of by using //  Eg: 2 // 5 = 2 |
| Unicode | Need to mention if it is unicode  Eg: print type(u"Hello") | By default it will take as unicode so no need to mention explicitly  Eg: print(type("Hello")) |
| Range Function | Xrange() function is to create the sequence of numbers | Used range() to do the same what xrange() function does so this function no longer in python 3 |
| Compatibility | Easy to upgrade python 3 from 2 | Not possible to downgrade to python 2 |

1. What is the difference between a tuple and a list in Python?

| List | Tuple |
| --- | --- |
| Mutable | Im-mutable |
| Enclosed in square brackets | Enclosed in parentheses |
| List = [1,2,3,4,5] | Tup = (1,2,3,4,5) |
| Items can be added, removed | Items cannot changed |
| Consume more memory | Consume less compared to List |
| Have many built-in methods  Eg: append(), pop(), remove(), count(), insert(), index(),... | Have not more built-in methods  Eg: count(), index() |

1. How do you create a dictionary in Python?

It is the collection of key value pairs. And will get the value through the key.

Duplicate keys are not allowed in each set.

Can store mixed data type values.

Eg:

dict = {1: "One", 2: 2, 3: 3.0, 4: "Four"}

print((dict[3]))

1. What is a function in Python and how do you define one?

It is a block of code and will execute once it is called.

Can pass the parameter to the function for dynamic results.

To create a function with the def keyword.

Mainly used to reduce redundancy.

Eg:

def get\_even\_numbers(lst):

even\_lst = []

for i in lst:

if i%2 == 0:

even\_lst.append(i)

print (even\_lst)

lst = [1,2,3,4,5,6,7,8,9,10,11,12,13,14]

get\_even\_numbers(lst)

1. What is object-oriented programming (OOP) and how does it relate to Python?

Python is also an object oriented programming language.

Everything in Python is objects and mostly everything has attributes and methods.

The OOP helps to develop applications using an object oriented approach.

The oops concepts mainly focus on reusable code.

OOPs concepts are

* Class
* Object
* Inheritance
* Polymorphism
* Data Abstraction
* Encapsulation

Create a class using the keyword “class”.

\_\_init\_\_() method is the first method will execute once the class called.

Method within the class has 4 indented spaces and body of the

Eg: Class SampleClassName

class NewFilee:

def \_\_init\_\_(self, name):

self.name = name

obj = NewFilee("Sample Class Creation")

print(obj.name)

1. How do you handle exceptions in Python?

To handle the exception using

* Try - Actual logic is placed to execute. Here, if there is no error then it will provide the actual result for logic
* Except - It will catch the error thrown from the try block and will display the error message provided
* Finally - This block will execute always irrespective of error thrown or not

Eg:

def test\_fun():

try:

fo = open("notexisitfile.txt", "r")

print(fo.read())

except IOError:

print ( "Input file not found" )

test\_fun()

1. How do you read and write files in Python?

To read:

First have to open the file with a path and specify the operations which are going to be done in the file.

Open file by using the open() with path and perform operations by default read only.

Operations:

r - read the file and it will throws error if file is not existed in provided path

w - Write and it will create new file if not existed in provided path

a - append the values in the existing file and it will create new file if not existed in provided path

x - it is to create a new file and will throws an error if already exists

Eg:

To read the file,

fo = open("C:\Vicknes\Training\Python\Prac\extfileio.txt", "r")

Or

fo = open("C:\Vicknes\Training\Python\Prac\extfileio.txt”)

print(fo.read())

To write in the existing file and will create new if not existed. It will overwrite the data in the file and not append the data.

fo = open("C:\Vicknes\Training\Python\Prac\extfileio.txt", "w")

fo.write("\n next line")

To write in the existing file and will create new if not existed. It appends the data in the file.

fo = open("C:\Vicknes\Training\Python\Prac\extfileio.txt", "a")

fo.write("\n next line")

1. How do you install and use external packages in Python?

External packages are downloaded and installed using pip which is a package manager.

After installing the package, use the package in the needed file by using import keyword.

Eg:

First check whether pip is already installed, for that use below

python -m pip --version

If already installed, then will proceed to install the packages or if not then have to download the pip and run they downloaded packed as below

python get-pip.py

To install the packages just use pip install and package name as below for install “pandas” packages

pip install pandas

Then import the package in the file needed and use the necessary function as below,

import pandas

Function dir(package) is to show all the attributes in the packages.

print(dir(pandas))

To use the attributes,

import pandas

print(pandas.\_\_version\_\_)

1. How do you use the "if" statement in Python to perform conditional execution?

a=10

b=5

c=15

if a> b:

print("a is bigger")

elif b > c:

print("b is bigger")

else:

print("c is bigger")