**In Class Programming-1**

Q) State differences between Python 2 and Python 3 version.

Ans) The major differences between python 2 and python 3 are:

1.The print function:

Python 2 print statement has been replaced by print ( ) function, meaning we have to wrap the object that we want to print in the parentheses, otherwise Python 3 would raise a syntax error.

In Python 2 print statement is written without parentheses.

Ex: for python 2:

Print ‘Hello world’

For Python 3:

Print (‘Hello world’)

2. Xrange

In Python 2 the usage of xrange ( ) is very popular for creating an iterable object ie. In a for loop or in list/set directory-comprehension.

In python 3 the range ( ) was implemented like the xrange ( ) and hence a dedicated xrange ( ) does not exist. When you type xrange ( ) in python 3 name error raises.

Ex: In python 2:

Print ‘\n\ntiming xrange ( )’

In python 3:

Print (‘\n timing range ( )’)

3.Handling Exceptions

In python 3 we have to use the “as” keyword for handling exceptions which we don’t have in python 2.

Ex: in python 2:

**try**:

let\_us\_cause\_a\_NameError

**except** NameError, err:

**print** err, '--> our error message'

in python 3:

**try**:

let\_us\_cause\_a\_NameError

**except** NameError **as** err:

**print** (err, '--> our error message')

4. next ( ) and .next ( ) function:

Python 2 has both next ( ) and .next( ) function but there is only a next( ) that remains in the python 3. Python 3 gives Attribute error when .next ( ) function is called.

Ex: in python 2:

my\_generator **=** (letter **for** letter **in** 'abcdefg')

next(my\_generator)

my\_generator**.**next()

In python 3:

my\_generator **=** (letter **for** letter **in** 'abcdefg')

next(my\_generator)

5. The \_future\_ module

Python 3 introduced some python 2 incompatible keywords and features that can be imported via the in-built \_future\_ module in python 2.

Ex: In python 3:

From \_ future\_ import division

6. Unicode

Python 2 has the ascII str ( ) types, separate unicode ( ) but no byte type.

Python 3 has unicode (utf-8) str ings and 2 byte classes namely: byte and bytearray s.

Ex: in python 2

**print** type (bytearray (b'bytearray oddly does exist though'))

in python 3:

**print** ('strings are now utf-8 \u03BCnico\u0394é!')