```
SERIES :

(1) Creation of Series.

Series :
 0    4
1    7
2   -5
3    3
dtype: int64
Values of Series :  [ 4  7 -5  3]
Indices of Series :  RangeIndex(start=0, stop=4, step=1)

(2) User-Based Indexing of Series.

Series2 :
 d    4
b    7
a   -5
c    3
dtype: int64
Values of Series2 :  [ 4  7 -5  3]
Indices of Series2 :  Index(['d', 'b', 'a', 'c'], dtype='object')
Index of User Based Index a : -5

Series2 after Inserting Element at Index d :
 d    6
b    7
a   -5
c    3
dtype: int64
Series with Specific Indices :
 c    3
a   -5
d    6
dtype: int64
```

```
Displaying Series2 with Condition :
 d    6
b    7
c    3
dtype: int64
Scalar Multiplication of Series2 :
 d    12
b    14
a   -10
c     6
dtype: int64
Exponential of Series2 :
 d     403.428793
b    1096.633158
a       0.006738
c      20.085537
dtype: float64

(3) Some More Series Implementations.

Series2 :
 Ohio      35000
Texas     71000
Oregon    16000
Utah       5000
dtype: int64
Series4 :
 California        NaN
Ohio          35000.0
Oregon        16000.0
Texas         71000.0
dtype: float64
Summation of Series3 & Series4 :
 California         NaN
Ohio           70000.0
Oregon         32000.0
Texas         142000.0
Utah               NaN
dtype: float64
```

```
Series4 names :
 state
California        NaN
Ohio          35000.0
Oregon        16000.0
Texas         71000.0
Name: population, dtype: float64

(4) Updating Series1 Indices :

Series1 with Updated Indices :
 Bob      4
Steve    7
Jeff    -5
Ryan     3
dtype: int64

DATAFRAME :

DataFrame1 :
     State  Year  Pop
0    Ohio  2000  1.5
1    Ohio  2001  1.7
2    Ohio  2002  3.6
3  Nevada  2001  2.4
4  Nevada  2002  2.9
5  Nevada  2003  3.2
Implementing head() func :
     State  Year  Pop
0    Ohio  2000  1.5
1    Ohio  2001  1.7
2    Ohio  2002  3.6
3  Nevada  2001  2.4
4  Nevada  2002  2.9
```

```
Reordering Columns of DataFrame1 :
    Year   State  Pop
0  2000    Ohio  1.5
1  2001    Ohio  1.7
2  2002    Ohio  3.6
3  2001  Nevada  2.4
4  2002  Nevada  2.9
5  2003  Nevada  3.2

DataFrame3 :
       year state  pop debt
one     NaN   NaN  NaN  NaN
two     NaN   NaN  NaN  NaN
three   NaN   NaN  NaN  NaN
four    NaN   NaN  NaN  NaN
five    NaN   NaN  NaN  NaN
six     NaN   NaN  NaN  NaN

Columns of DataFrame3 :  Index(['year', 'state', 'pop', 'debt'], dtype='object')

Accessing with Indexing :
 one      NaN
two      NaN
three    NaN
four     NaN
five     NaN
six      NaN
Name: state, dtype: object
Accessing with Attribute :
 one      NaN
two      NaN
three    NaN
four     NaN
five     NaN
six      NaN
Name: year, dtype: object
Retrieving Rows with loc() :
 year     NaN
state    NaN
pop      NaN
debt     NaN
Name: three, dtype: object
```

```
Inserting a Common Element in Every Rows :
       year state  pop  debt
one     NaN   NaN  NaN  16.5
two     NaN   NaN  NaN  16.5
three   NaN   NaN  NaN  16.5
four    NaN   NaN  NaN  16.5
five    NaN   NaN  NaN  16.5
six     NaN   NaN  NaN  16.5
Arranging The Inserted Values :
       year state  pop  debt
one     NaN   NaN  NaN   0.0
two     NaN   NaN  NaN   1.0
three   NaN   NaN  NaN   2.0
four    NaN   NaN  NaN   3.0
five    NaN   NaN  NaN   4.0
six     NaN   NaN  NaN   5.0
      year state  pop  debt
one    NaN   NaN  NaN   NaN
two    NaN   NaN  NaN  -1.2
three  NaN   NaN  NaN   NaN
four   NaN   NaN  NaN  -1.5
five   NaN   NaN  NaN  -1.7
six    NaN   NaN  NaN   NaN
      year state  pop  debt  eastern
one    NaN   NaN  NaN   NaN    False
two    NaN   NaN  NaN  -1.2    False
three  NaN   NaN  NaN   NaN    False
four   NaN   NaN  NaN  -1.5    False
five   NaN   NaN  NaN  -1.7    False
six    NaN   NaN  NaN   NaN    False
Index(['year', 'state', 'pop', 'debt'], dtype='object')
Dictionary to DataFrame :
      Nevada  Ohio
2001     2.4   1.7
2002     2.9   3.6
2000     NaN   1.5
```

```
Transposing DataFrame4 :
         2001  2002  2000
Nevada    2.4   2.9   NaN
Ohio      1.7   3.6   1.5
Slicing The DataFrame :        Ohio  Nevada
2001   1.7     2.4
2002   3.6     2.9

INDEX OBJECTS :

Indices of New Series :  Index(['a', 'b', 'c'], dtype='object')
Slicing Indices :  Index(['b', 'c'], dtype='object')
Sharing Index Objects :  Index([0, 1, 2], dtype='int32')
Index DataFrame2 :
 0    1.5
1   -2.5
2    0.0
dtype: float64
Checking for Indices :  True
```