Answer key of Comprehensive examination for Programming for Analytics

Q1. What will be the output of the following Python code

In [ ]:
```
y = 5
x = y ^ 5 # ^ is the XOR operator
x # Since both of its operands are same XOR will return 0 and will be prin
```

Out[42]:
```
0
```

Q2. Which of the following code snippet will print “Hi”

In [ ]:
```
# Code snippet 1
if not 3 >= 0:
  print("Hi")

# 3 is greater than 0, so >= operator will return True.
# This True is then passed to not operator returning False to if condition
```

In [ ]:
```
# Code snippet 2
if (not 3) >= 0:
  print("Hi")
# 3 is considered True, not True is False
# False >= 0 because of equality operator it returns True to if condition
```

```
Hi
```

Q3. What will be the output of the following Python code

In [ ]:
```
x = -1 and -1
x

# Both operands to the and operator is same (-1) and non-zero, hence it wi
```

Out[39]:
```
-1
```

Q4. Write True/False as your answer

- 4.1. Python does not support method overloading - True
- 4.2. In Python, everything is an object - True
- 4.3. The ‘assert’ keyword is primarily used for debugging in Python - True
- 4.4. Numpy’s histogram() function is used to draw a histogram - False (It only calculates data for a histogram)

Q5. In Pandas, which function displays basic statistics (mean, median, etc.) of numeric columns in a DataFrame?

- A) describe()
- B) summary()
- C) stats()
- D) analyze()

Answer: A) describe()

Q6. Briefly explain 2 advantages of the display() method over the print() method.

Answer: Printing a data frame through display() enhances readability and provides interactivity as compared to print() funtion

Q7. Match the following

1. Seaborn - I. is a superset of the Matplotlib library
2. Plotly - F. provides interactive visualizations
3. Numpy - A. is a dependency of Pandas
4. Histogram - D. is a one-dimensional visualization
5. Operator overloading - H. is an example of polymorphism
6. Scikit-learn - B. Python library used for modeling
7. Scipy - G. builds on top of Numpy
8. `__str__` - J. Redefining it would be known as method overriding.
9. self - E. An instance of the class
10. _var1 - C. Would be considered a protected data member

Answer: 1I, 2F, 3A, 4D, 5H, 6B, 7G, 8J, 9E, 10C

Q8. What is the purpose of using value_counts() in Pandas?

- A) To display summary statistics of a DataFrame
- B) To count the number of unique values in a column
- C) To sort the DataFrame by column values
- D) To aggregate and group data in a DataFrame by specific columns

Answer: B) To count the number of unique values in a column

Q9. Which Pandas method is used to select a single column from a DataFrame?

- A) get_column()
- B) select()
- C) column()
- D) [ ]

Answer: D) [ ]

Q10. Which Pandas method drops rows with missing values in a DataFrame?

- A) remove_missing()
- B) drop_null()
- C) dropna()
- D) clean_data()

Answer: C) dropna()

Q11.

- Q11.1 Write the output of line no. 10
- Q11.2 Write the output of line no. 13
- Q11.3 Write the output of line no. 16
- Q11.4 Write the output of line no. 19
- Q11.5 Write the purpose of axis argument (in which manner it performs operations)?

Answer:

- axis argument performs operation either row-wise or column-wise
- By default axis is 0 which performs operation row-wise
- axis argument when given 1 performs operations column-wise

In [ ]:

```python
import pandas as pd

data = {
    'A': [1, 2, 3],
    'B': [4, 5, 6],
    'C': [7, 8, 9]
}

df = pd.DataFrame(data)
print(df)

sum = df.sum(axis=0)
print("\nSum\n", sum)

sum = df.sum(axis=1)
print("\nSum\n", sum)

sum = df.sum()
print("\nSum\n", sum)
```

```
   A  B  C
0  1  4  7
1  2  5  8
2  3  6  9

Sum
 A     6
B    15
C    24
dtype: int64

Sum
 0    12
1    15
2    18
dtype: int64

Sum
 A     6
B    15
C    24
dtype: int64
```

Q12. Briefly mention the purpose of following Python functions

- 12.1. pd.concat() : Can be used to concatenate two data frames
- 12.2. np.cumsum() : Performs the cumulative summation
- 12.3. add_legend() : Adds legend to a plot to represent data point information through color coding
- 12.4. head() : Prints specific number of top rows of a DataFrame
- 12.5. fillna() : Filling missing values with a specific value

Q13. Briefly (1-3 lines only) explain following Python concepts

- 13.1. Pickling and What are pickles? - Pickling is the process of saving python objects such as variables and methods into binary text files for later usage.
- 13.2. Dunder methods - Dunder methods define built-in behaviors such as print function and and **init** for initialization of an object.
- 13.3. A nullary function - A nullary function, accepts no arguments.
- 13.4. Access modifiers - Access modifiers control the visibility and accessibility of class attributes and methods in Python. E.g. public, private and protected
- 13.5. Operator overloading - Allows you to provide custom behaviors for built-in operators like +, -, *,

Q14.

- Q14.1 self and other are two methods defined inside Point class in line no. 6 (True/False) - False (self and other are objects instance and not methods)
- Q14.2 Line no. 8 overloads the '>' operator (True/False) - False (it overloads == operator)
- Q14.3 Line no. 11 and 12 creates two Point type methods (True/False) - False (It creates two objects of Point type)
- Q14.4 Line no. 14 ***adds*** two Point objects using the overloaded **+** operator
- Q14.5 Line no. 17 tests **equality** between two point type objects using == operator

In [ ]:

```
# Comprehend the Python code written below and write True/False or fill in
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)
    def __eq__(self, other):
        return self.x == other.x and self.y == other.y

point1 = Point(3, 4)
point2 = Point(5, 6)

result = point1 + point2
print("Addition result: (x =", result.x, ", y =", result.y, ")")

equal = point1 == point2
print("Equality:", equal)
```

```
Addition result: (x = 8 , y = 10 )
Equality: False
```

Q15. Briefly (1-3 lines) explain each line of Python code snippet given below with their functionality/output generated

In [ ]:

```
import pandas as pd # Imports pandas module with alias pd
import requests as re # Imports requests module with alias re
r = re.get('https://www.google.co.in') # Will request to get the html page
print(r) # Generates the status of web request
print(r.text) # Contains the source code of html page loaded from the URL
```

```
<Response [200]>
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang
="en"><head><meta content="Search the world's information, including webpag
es, images, videos and more. Google has many special features to help you f
ind exactly what you're looking for." name="description"><meta content="noo
dp" name="robots"><meta content="text/html; charset=UTF-8" http-equiv="Cont
ent-Type"><meta content="/logos/doodles/2023/seasonal-holidays-2023-6753651
837110165-law.gif" itemprop="image"><meta content="Seasonal Holidays 2023"
property="twitter:title"><meta content="Happy Holidays! #GoogleDoodle" prop
erty="twitter:description"><meta content="Happy Holidays! #GoogleDoodle" pr
operty="og:description"><meta content="summary_large_image" property="twitt
er:card"><meta content="@GoogleDoodles" property="twitter:site"><meta conte
nt="https://www.google.com/logos/doodles/2023/seasonal-holidays-2023-675365
1837110165-2xa.gif" property="twitter:image"><meta content="https://www.goo
gle.com/logos/doodles/2023/seasonal-holidays-2023-6753651837110165-2xa.gif"
property="og:image"><meta content="1000" property="og:image:width"><meta co
ntent="400" property="og:image:height"><meta content="https://www.google.co
m/logos/doodles/2023/seasonal-holidays-2023-6753651837110165-2xa.gif" prope
rty="og:url"><meta content="video.other" property="og:type"><title>Google</
```

Q16. What below Python code will do?

In [ ]:

```
a = 0; b = 5 # One liner assignment of variables
b,a = (a,b) # a and b will swap their values
res = a/b # Division by 0
print(res)

# Will generate the ZeroDivisionError
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-35-b8cabcedab3c> in <cell line: 3>()
      1 a = 0; b = 5 # One liner assignment of variables
      2 b,a = (a,b) # a and b will swap their values
----> 3 res = a/b # Division by 0
      4 print(res)
      5

ZeroDivisionError: division by zero
```

Q17. Given a file with name “some_text.txt” in the working directory with content

```
#-TEXT
```

Three file handling operations (o1, o2 and o3) have been defined below.

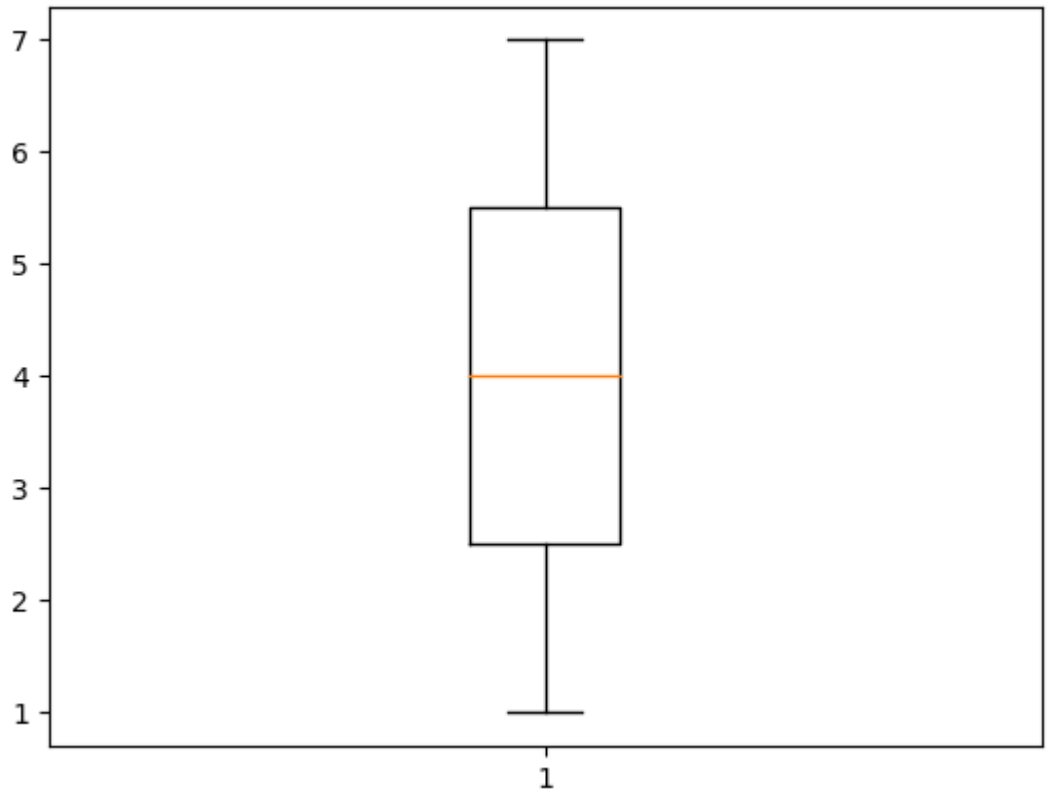
In [ ]:

```python
def o1():
  file = open("some_text.txt", "r")
  print(file.read())

def o2():
  file = open("some_text.txt", "a")
  file.write("-DELETE-ALL")
  file.close()

def o3():
  file = open("some_text.txt", "w")
  file.write("TEXT-#")
  file.close()

o1()
o2()
o1()
o3()
o1()
o3()
o1()
o2()
o1()

# Print the output of the above Python code (5 print statements as execute
```

```
#-TEXT
#-TEXT-DELETE-ALL
TEXT-#
TEXT-#
TEXT-#-DELETE-ALL
```

Q18. Draw a neatly labelled visualization to pictorially represent the concept of a boxplot and its IQR method to identify outliers. Make proper markings of axes and other values. Show median and other quartiles for the given variable ‘x’ as shown below in Python code. Do not write theory, only use proper visualization to explain the concept.

In [ ]:
```
import matplotlib.pyplot as plt
x = [1,2,3,4,5,6,7]
plt.boxplot(x)
plt.show()
```



Q19. What below Python code will do?

In [ ]:
```
x = "Hello"
assert x == "Hi", "This is not Hello"
# The code will raise Assertion error since the value of x is "Hello" and
```

Q20. Given a file with the name “some_text.txt” in the working directory with content

```
#-TEXT
```

In [ ]:

```python
#What the following Python code will do?
def o4():
    file = open("some_text.txt", "w")
    file.write("-DELETE-ALL")
    print(file.read())
    file.close()

o4()
# The code block will return exception as file was opened in write mode an
```

```
---------------------------------------------------------------------------
UnsupportedOperation                      Traceback (most recent call last)
<ipython-input-34-c4f12aa4943f> in <cell line: 8>()
      6     file.close()
      7
----> 8 o4()
      9 # The code block will return exception as file was opened in write mo
de and a read operation was requested in line no. 5

<ipython-input-34-c4f12aa4943f> in o4()
      3     file = open("some_text.txt", "w")
      4     file.write("-DELETE-ALL")
----> 5     print(file.read())
      6     file.close()
      7

UnsupportedOperation: not readable
```

There are several modes in which files can be opened such as:

- r - read mode
- w - write mode
- a - append mode
- `+` as as suffix will allow updation along with the read or write or append mode as per the cursor/handle position used by the corresponding open mode
- `b` as a suffix will allow file open operation in binary mode for byte stream data written inside a file.