Introduction to Big Data - Practice 1 (Python Programming)

For each question, make sure you not only just “write down” Python codes but also “explain the answer with your own language”. All answers without explanation will not be accepted.

Problem

< Question 1 >

Write down Python codes in Google Colab, run them, and explain with your own words how it works.

(1-1, 5 pts) Write down Python codes and explain the result. From this, what can we notice about the variable? You must mention the key concept of “variable”.

Sample Codes

```
a = 123
A = 234
print(a, id(a))
print(A, id(A))
```

(1-2, 5 pts) Why 123, 123.0 and ‘123’ are different in programming? Write down Python code that shows their difference and explain.

(1-3, 5 pts) Create a variable called *myVerse* that stores Bible verse from Matthew 1: 20-25 as shown below.

Expected Results

```
>>> myVerse
But after he had considered this, an angel of the Lord appeared to
him in a dream and said, "Joseph son of David, do not be afraid to
take Mary home as your wife, because what is conceived in her is from
the Holy Spirit. She will give birth to a son, and you are to give
him the name Jesus, because he will save his people from their sins."
All this took place to fulfill what the Lord had said through the
prophet: "The virgin will be with child and will give birth to a son,
and they will call him Immanuel"--which means, "God with us." When
Joseph woke up, he did what the angel of the Lord had commanded him
and took Mary home as his wife. But he had no union with her until
she gave birth to a son. And he gave him the name Jesus.
```

(1-4, 5 pts) In *myVerse*, how many times do Jesus and Lord appear? Write Python codes that present the expected results as shown below.

Expected Results

```
>>> ???
Jesus appears 2 times and Lord appears 3 times.
```

(1-5, 5 pts) From *myVerse*, write down Python codes that extract the word Jesus as shown below.

Expected Results
>>> ??? Jesus

(1-6, 5 pts) Write down following two versions of Python codes, present the result and explain why they show the different results.

Sample Codes	
(a)	(b)

`age = 16`
`float(age)`
`type(age)`
	`age = 16`
	`age = float(age)`
	`type(age)`

< Question 2 >

Given the results of Python codes, answer the following questions.

(2-1, 10 pts) Why is this code not working? Explain and suggest the solution to fix this error.

Results
>>> score ---------- NameError Traceback (most recent call last) <ipython-input-29-d2d780e36333> in <cell line: 1>() ----> 1 score NameError: name 'score' is not defined

(2-2, 10 pts) Why is this code not working? Explain and suggest the solution to fix this error.

Results
>>> 2_boys = 'James, Bruce' File "<ipython-input-30-e1f382b43259>", line 1 2_boys = 'James, Bruce' ^ SyntaxError: invalid decimal literal

(2-3, 10 pts) Why is this code not working? Explain and suggest the solution to fix this error.

Results
>>> 1 + '1' ---------- TypeError Traceback (most recent call last) <ipython-input-49-7ff5cb60d31b> in <cell line: 1>() ----> 1 1 + '1' TypeError: unsupported operand type(s) for +: 'int' and 'str'

(2-4, 10 pts) Why is this code not working? Explain and suggest the solution to fix this error.

Results

```
>>> import numpi as np
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-31-119be8593034> in <cell line: 1>()
----> 1 import numpi as np
      2
      3 numpi

ModuleNotFoundError: No module named 'numpi'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

< Question 3 >

Write down Python codes in Google Colab, run them, and explain with your own words how it works.

(3-1, 10 pts) Define a function called number() that separates even and odd numbers (NOTE: Use if and else statement). As shown below, run a simple code that shows the same results.

Expected Results

`>>> number(10)`
`Even : 10`
`>>> number(11)`
`Odd : 11`

(3-2, 15 pts) Create a class called *Calculator*, which contains five methods called "Add", "Minus", "Multiply", "Division", and "Square". (*Add: adds two values, Minus: substract two values, Multiply: multiply two values, Division: Divide first value by second value, Square: Square the second value - ex. x^2)

Expected Results
`>>> Calculator` `main  .Calculator`

(3-3, 5 pts) With *calculator*, create an instance called "a" and receive 10 and 5 as an input. With "a", do the following simple calculation to check whether it is made properly.

Expected Results

`>>> a.Add()`
`???`
`>>> a.Minus()`
`???`
`>>> a.Multiply()`
`???`
`>>> a.Division()`
`???`
`>>> a.Square()`
`???`