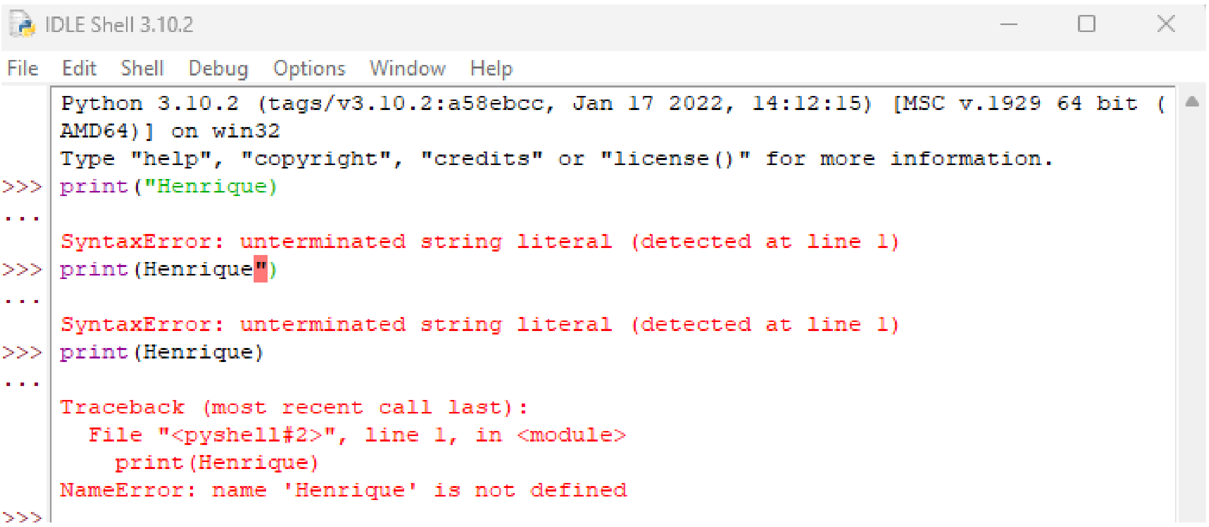
Part 1: LEARN FROM YOUR MISTAKES!

a.If you are trying to print your name, what happens if you leave out one of the quotation marks or both, and why?

In the case of leave out one of the quotation, Python throws an SystaxError message.

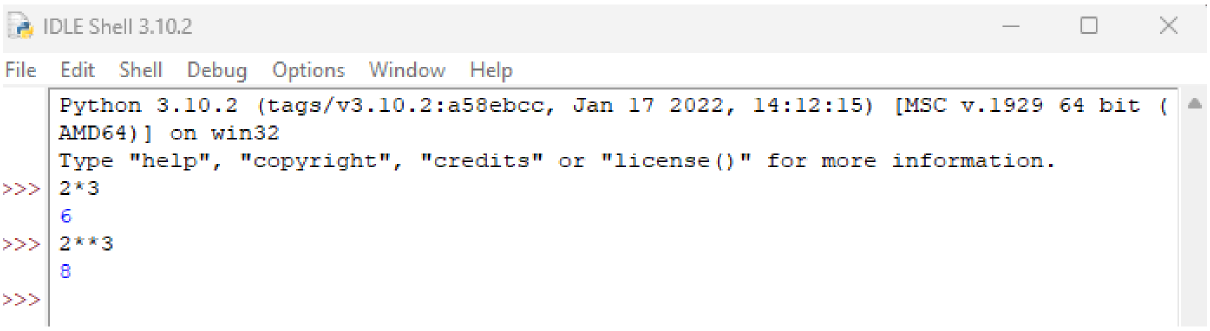
In the case of leaving out both, Python understands the input as a variable. Because this possible variable is not declared, Python throws a NameError message.



```
    Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
    AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Henrique)
...
    SyntaxError: unterminated string literal (detected at line 1)
>>> print(Henrique")
...
    SyntaxError: unterminated string literal (detected at line 1)
>>> print(Henrique)
...
    Traceback (most recent call last):
      File "<pyshell#2>", line 1, in <module>
        print(Henrique)
    NameError: name 'Henrique' is not defined
>>>
```

b.What is the difference between * and ** operators in Python? Explain with the help of an example.

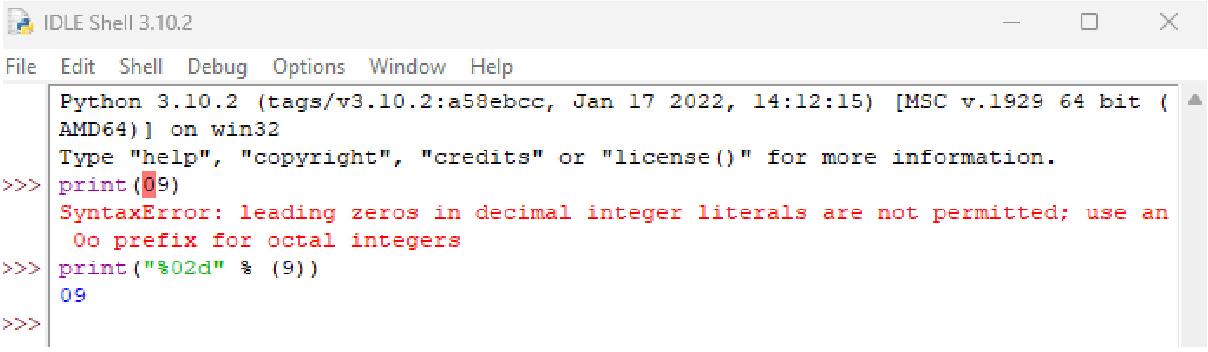
The * operator is responsible for multiplications, and the ** operator is responsible for exponentiation.



```
    Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
    AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>> 2*3
    6
>>> 2**3
    8
>>>
```

c.In Python, is it possible to display an integer like 09? Justify your answer.

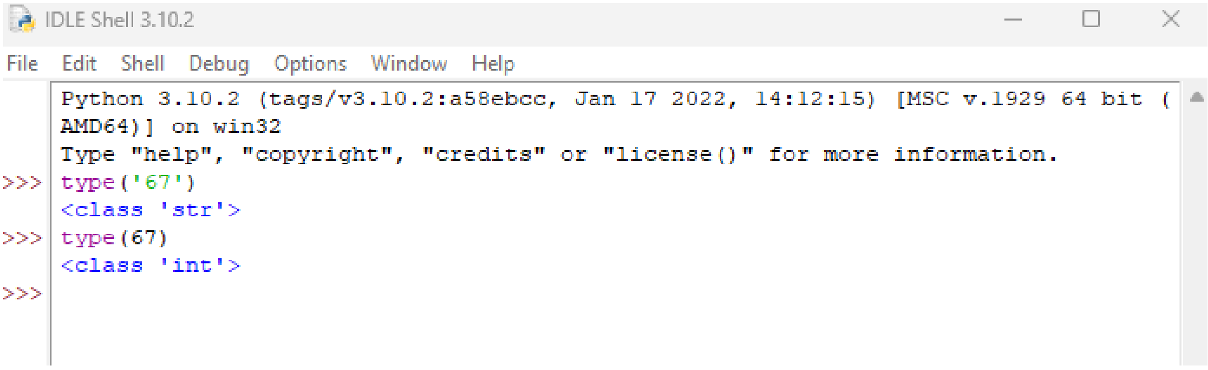
No, because it has a leading zero. If you need to display the value “09” you should pass it as a string.



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print(09)
SyntaxError: leading zeros in decimal integer literals are not permitted; use an
 0o prefix for octal integers
>>> print("%02d" % (9))
09
>>>
```

d.Run the commands type(‘67’) and type(67). What is the difference in the output and why?

Type(‘67’) shows that the input is a string and type(67) show that the input is an integer. That happens because, although both are numbers, ‘67’ is interpreted by Python as an array of characters, and 67 as a number.



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> type('67')
<class 'str'>
>>> type(67)
<class 'int'>
>>>
```

Part2:

I have created a class using the Object-Oriented Programming approach to handle the operations required for this assessment. This class named “AssessmentPart2” contains 7 properties, one to each information needed to perform each task, i.e. age, city, country, continent, startDateExam, endDateExam and temperature. I also made in this class four functions, each one responsible for the message and logic behind each task, encapsulating the rules for each message to be displayed. After instantiating the class passing the correct values, the only action needed is to call the class functions to obtain the values to be displayed. In an enterprise environment I would have made this different, but because this is not the case, hopefully this will be enough to cover what is demanded in this assessment.

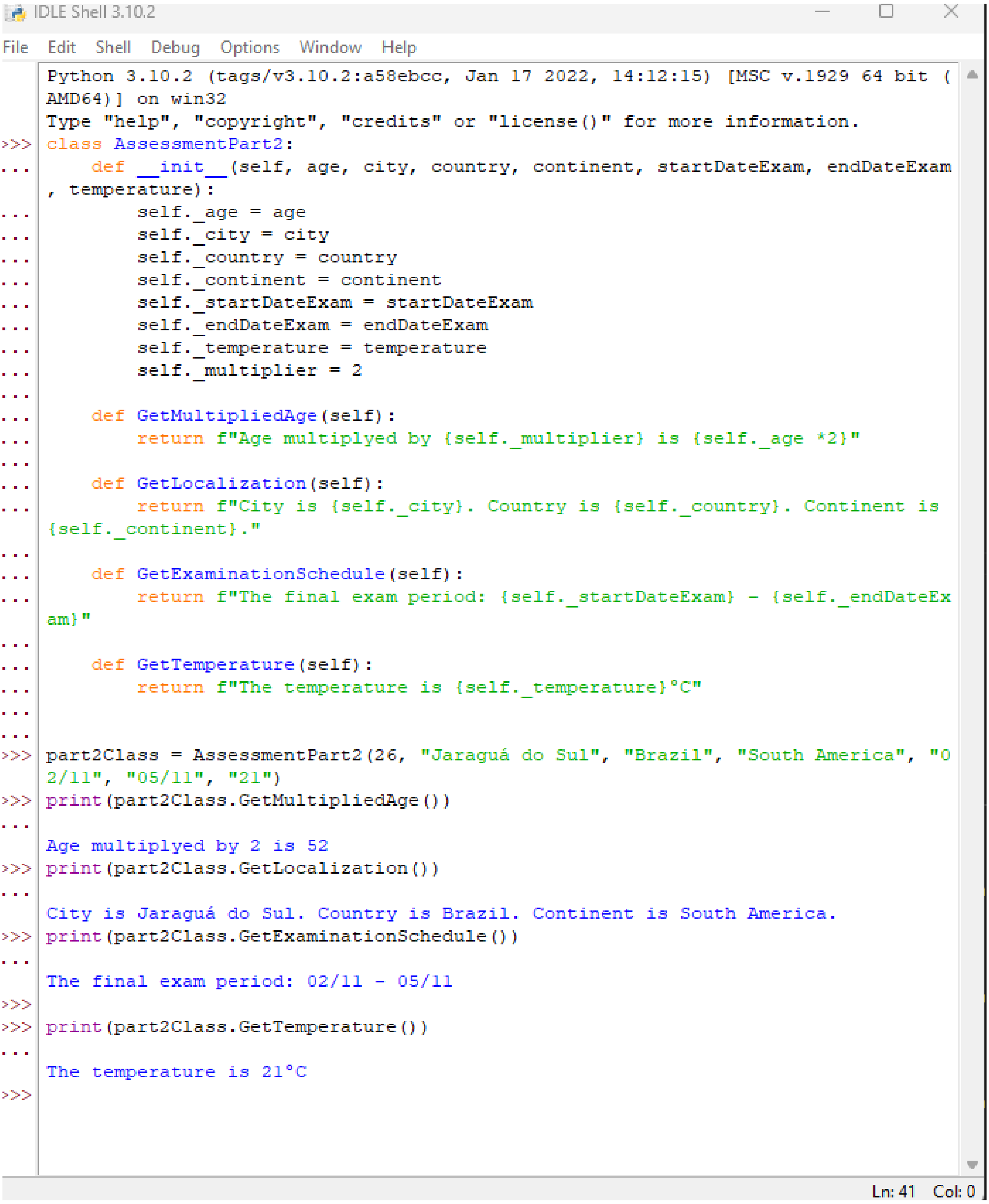
I already develop using C#, so I have decided to use this approach to handle this assessment to give me the opportunity to apply concepts that I have studied previously by myself, but using Python. I have learned with this approach: how to create properties in Python; how to create a class in Python; how to create a constructor method in Python; how to create functions in Python; how to interpolate strings in Python; what is the “self” word in Python.

REFERENCES

Object-oriented programming. (2023, September 07). In *Wikipedia*. https://en.wikipedia.org/wiki/Object-oriented_programming

Component statements (2023, September 07). In *Python. https://docs.python.org/3/reference/compound_stmts.html#class-definitions*

Defining Functions (2023, September 07). In *Python. https://docs.python.org/3/tutorial/controlflow.html#defining-functions*



```
    Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
    AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>> class AssessmentPart2:
...     def __init__(self, age, city, country, continent, startDateExam, endDateExam
    , temperature):
...         self._age = age
...         self._city = city
...         self._country = country
...         self._continent = continent
...         self._startDateExam = startDateExam
...         self._endDateExam = endDateExam
...         self._temperature = temperature
...         self._multiplier = 2
...
...     def GetMultipliedAge(self):
...         return f"Age multiplyed by {self._multiplier} is {self._age *2}"
...
...     def GetLocalization(self):
...         return f"City is {self._city}. Country is {self._country}. Continent is
    {self._continent}."
...
...     def GetExaminationSchedule(self):
...         return f"The final exam period: {self._startDateExam} - {self._endDateEx
    am}"
...
...     def GetTemperature(self):
...         return f"The temperature is {self._temperature}°C"
...
...
>>> part2Class = AssessmentPart2(26, "Jaraguá do Sul", "Brazil", "South America", "0
    2/11", "05/11", "21")
>>> print(part2Class.GetMultipliedAge())
...
    Age multiplyed by 2 is 52
>>> print(part2Class.GetLocalization())
...
    City is Jaraguá do Sul. Country is Brazil. Continent is South America.
>>> print(part2Class.GetExaminationSchedule())
...
    The final exam period: 02/11 - 05/11
>>>
>>> print(part2Class.GetTemperature())
...
    The temperature is 21°C
>>>
```

```
class AssessmentPart2:
    def __init__(self, age, city, country, continent, startDateExam, endDateExam,
temperature):
        self._age = age
        self._city = city
        self._country = country
        self._continent = continent
        self._startDateExam = startDateExam
        self._endDateExam = endDateExam
        self._temperature = temperature
        self._multiplier = 2

    def GetMultipliedAge(self):
        return f"Age multiplyed by {self._multiplier} is {self._age *2}"

    def GetLocalization(self):
        return f"City is {self._city}. Country is {self._country}. Continent is {self._continent}."

    def GetExaminationSchedule(self):
        return f"The final exam period: {self._startDateExam} - {self._endDateExam}"

    def GetTemperature(self):
        return f"The temperature is {self._temperature}°C"
```

```
part2Class = AssessmentPart2(26, "Jaraguá do Sul", "Brazil", "South America", "02/11",
"05/11", "21")
print(part2Class.GetMultipliedAge())
print(part2Class.GetLocalization())
print(part2Class.GetExaminationSchedule())
print(part2Class.GetTemperature())
```