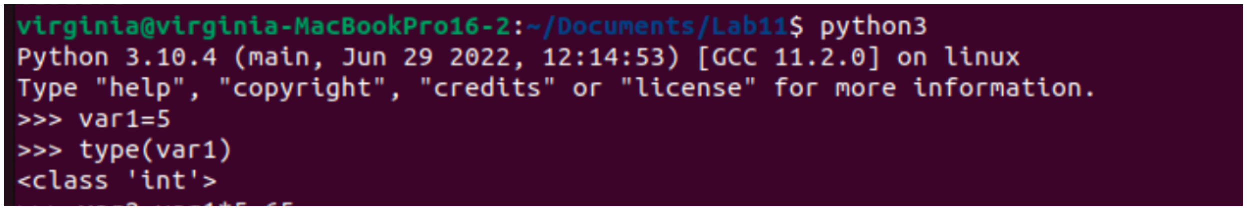
Lab Instructions:

2.1. Type in var1 = 5 and press ENTER.
2.2.To check the data type of a, type in type(a) and press ENTER.

2.2.1. What is the data type of the variable 'var1' ? State your answer in your own words and paste a screenshot of your output.

The data type of var1 is an integer.



```
virginia@virginia-MacBookPro16-2:~/Documents/Lab11$ python3
Python 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> var1=5
>>> type(var1)
<class 'int'>
```

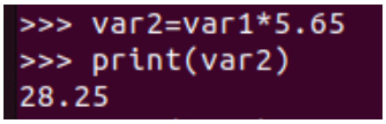
2.2.2.Notice that the interpreter knows the data type without us specifying it. What does this indicate about the Type Binding that Python uses? State your answer in <1 line.

Python uses dynamic type binding where it decides the variable's data type during runtime.

2.3. Let us introduce a second variable 'var2' now. Assign value to var2 by typing var2 = var1*5.65 and press ENTER. We can check the value assigned to var2 by typing print(var2)

2.3.1.What is the value assigned to variable 'var2' ? State your answer in your own words and paste a screenshot of your output.

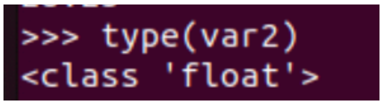
The assigned value is 28.25.



```
>>> var2=var1*5.65
>>> print(var2)
28.25
```

2.3.2.What is the data type of variable 'var2' ? Use the command from 2.2 to check the data type. State your answer in your own words and paste a screenshot of your output.

The data type of var2 is a float.



```
>>> type(var2)
<class 'float'>
```

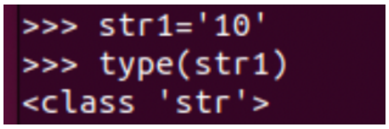
2.3.3.Notice that 'var1' and 'var2' have different data types. What characteristic of data type conversions in Python does this behavior indicate? State your answer in <1 line.

Python uses coercion to provide a compatible data type.

2.4.Let us introduce a third variable 'str1' now. This variable will hold a string value. Assign value to str1 by typing str1 = '10' and press ENTER.

2.4.1.What is the data type of variable 'str1' ? Use the command from 2.2 to check the data type. State your answer in your own words and paste a screenshot of your output.

The data type of str1 is a class that takes a string as an array of chars.

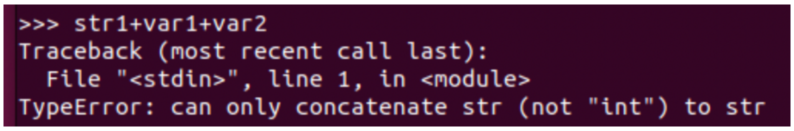


```
>>> str1='10'
>>> type(str1)
<class 'str'>
```

2.5.Let us try to perform an arithmetic addition operation adding str1, var1, and var2. Type in str1 + var1 + var2 and press ENTER.

2.5.1.What is the output that you observe? State your answer in your own words and paste a screenshot of your output.

str1+var1+var2 produces no answer. Instead, it outputs a type error that it cannot concatenate a string to an integer.



```
>>> str1+var1+var2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

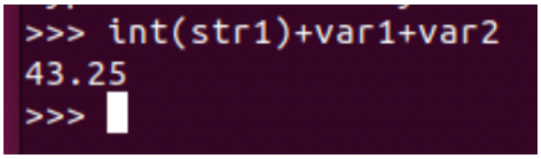
2.5.2.Which activity performed by the interpreter results in this output?

The interpreter’s type checker detected the error. It checked the operands data types to see if they are compatible. In this case, it was not.

2.6.Python also supports explicit typecasting where we can type cast one data type to another. To typecast an int ‘var1’ to a string, we can use (str)int1 or “int1”. Let us try to make the following changes to the arithmetic edition expression we were using in 2.5. Type in (int)str1 + var1 + var2 and press ENTER.

2.6.1. What is the output that you observe? State your answer in your own words and paste a screenshot of your output.

The output displays a value of 43.25.



```
>>> int(str1)+var1+var2
43.25
>>> 
```

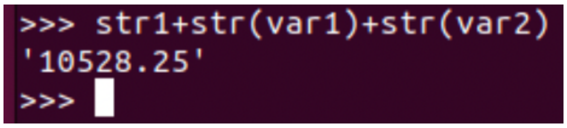
2.6.2.Why did you not see an error this time?

There was no error this time because the interpreter can read and understand when a variable has been typecasted. Here, str1 was no longer considered as a class but as an integer instead. Since all three variables have compatible data types, an answer was produced by adding the values.

2.7. Now type in str1 + str(var1) + str(var2) and press ENTER.

2.7.1. What is the output that you observe? State your answer in your own words and paste a screenshot of your output.

The output produces the answer ’10528.25’ in string format instead of a value that is an integer data type.



```
>>> str1+str(var1)+str(var2)
'10528.25'
>>> 
```

2.7.2.What operation does this expression seem to be doing? State your answer in <1 line.

This expression is concatenating the strings from each of the three variables (’10’+’5’+’28.25’).