Question 2: What is the difference between Interactive Mode and Script Mode?

Interactive Mode is pretty much coding in the shell, where you type individual Python statements into the shell using the characters “>>>” followed by your code. However, in Script Mode you’re able to read all of your statements in your .py file line by line, and then execute them. Interactive mode is better when only typing a few lines to test some stuff out, while script mode is better when your programs are much longer and more complicated.

Question 3: Why must all programs start with a docstring, and what should it contain?

All programs should start with a docstring as they are useful for explaining the main features of your code. The docstring at the top of your program, also known as your header comment, should contain information about the author, the date the program was written, as well as a brief description of what the program does.

Question 4: Why include code comments if the Python interpreter ignores them?

Comments make it easier for people reading/modifying the code to understand what the code does, so they don’t have to go through the work of having to spend time reading and trying to recognize what each piece of the program does. Code comments are especially useful if you’re collaborating with other programmers to work on one project together, as you won’t need to explain to your peers what each piece does.

Question 5: What is the difference between a legal variable name and a good variable name?

Legal variable names include any variable names that do not result in syntax errors, however good variable names are much more descriptive and provide information as to what the variable stores.

Question 6: Which of the following variable names would give a syntax error (and why), and which would be considered poor style (and why)?

- **y**
    - This would not give a syntax error, yet it would be considered poor style as the name does not provide much information as to what the variable does.
- **2cool**
    - This would not give a syntax error, yet it would be considered poor style as the name does not provide much information as to what the variable does.
- **thehighestscore**
    - This would not give a syntax error, and it does a fairly good job at explaining what the variable does.
- **Student_Average**
    - This would not give a syntax error, and it does a fairly good job at explaining what the variable does.

- **term&sumative**
    - This is an illegal variable name, as the & key is used as an operator in Python, meaning if you were to create a variable with this name, it would not work. This is also a poor variable name, as calling it term_sumative would be a better idea.

Question 7: Python supports three data types. List them, and give an example value that each might store.

- Integer
    - Integers in Python represent whole numbers, with no decimal points. Examples of integers could be numbers like 5, 3, or -17.
- String
    - Strings in Python represent sequences of characters, that can be either numbers, letters, or symbols. Examples of strings good be words like “hello”, numbers like “89” or symbols like “!$!$!$!”.
- Float
    - Float values in Python represent real numbers that have decimals, used for values that need more precision. Examples of these values can be numbers like 2.58, 8.39, or -56.91.

Question 8: Your friend is trying to write a Python program, help him fix the errors:

```
17 = Age
print (‘My age is’, ‘Age')
```

To start, in line 1, he wrote 17 = Age instead of Age = 17, which doesn’t assign a variable at all and commonly results in an error. This can be corrected by fixing the first line to read Age = 17. The second line is wrong for a few different reasons. To start, my friend used ` instead of “ or ‘, which doesn’t change the text into a string. There is also no variable named Age, leading to another error. Additionally, you need to put the brackets right after the print statement for better formatting, and the Age variable should not be encased in and ‘ otherwise it will just print “Age” instead of the actual Age value.