1. **Code Question Brix** —
   - Pearson will give you MPL questions that should be arranged in a Nemo book structure
   - Pearson will give you the chapter names and module names
   - The example below is chapter 7 from Gaddis Java. On Nemo you can look up GUID 221214e3-7450-46e4-8e0c-179e9715aed7 Chapter 7 to see how it is arranged.

Example:

Chapter 7
7.1
20690
20691
20692
20759
60123
60124
20695
20696
20699
20701
20706
20672
7.2
None
7.3
20650
20659
20662
7.4
20714
20719
7.5
None
7.6
None
7.7
None
7.8
20765
7.9
21030
21025
21027
20795
21033

21034
21035
21112
7.10
21032
7.11
20711
20712
7.12
None
7.13
20824
20823
20825
20826
20827
20832
20834
End of Chapter
71116
20749

2. **Simple Code Question Brix** —

- Your Pearson contact will tell you which questions are Simple Code Question Brix. In general
  - Simple Code Questions have only one line as an answer
  - Are created by the vendor (on MPL they will have exercises numbers that don't begin with 00000)
  - Are not placed at the end of every section
  - Will have a title (for example, Checkpoint 4.5)
- Please note it is not your responsibility to decide which question is Simple Code, so feel free to double check and confirm with your Pearson contact that it is indeed Simple Code
- Your Pearson contact will tell you exactly which questions should be placed in which chapter. For example:

7	Checkpoint 7.15
7	Checkpoint 7.16
7	Checkpoint 7.17
7	Checkpoint 7.18
7	Checkpoint 7.19
7	Checkpoint 7.20

7	Checkpoint 7.21
7	Checkpoint 7.22
7	Checkpoint 7.23
7	Checkpoint 7.24
7	Checkpoint 7.26
7	Checkpoint 7.27
7	Checkpoint 7.29
7	Checkpoint 7.30
7	Checkpoint 7.35
7	Checkpoint 7.36
7	Checkpoint 7.37
7	Checkpoint 7.38
7	Checkpoint 7.47
7	Checkpoint 7.48
7	Checkpoint 7.49
7	Checkpoint 7.50

3. Live Example Brix

- You will be given the question and correct answer.
- For example, the below code is in this GUID 6177dc13-d693-40b0-ab3f-1ffc0e99c24e
- Notice how some instructions such as “type the code” etc. was removed. This is because the code is already displayed to the student. You will need to use your discretion here.

Live Code Example 3.5

The code below asks the user how many steps they have walked in a day. Feedback is given based on the number of steps, which are defined as constants. Type the code into the interface below and fill in the two commented lines with the missing code to display the expected output. Type the code into the interface below and correct the errors to display the expected output.

Sample Expected Output

How many steps did you walk today? 9000
Good work! Just a few more left to go.

Code to Display and for the Student to Type In

```
# Constants for step levels
A_STEPS = 10000
B_STEPS = 8000
C_STEPS = 6000
```

```
D_STEPS = 4000

numsteps = int(input('How many steps did you walk today? '))

if numsteps >= A_STEPS:
   print('Great job!')
# Fill in the missing code here
   print('Good work! Just a few more left to go.')
elif numsteps >= C_STEPS:
   print('Nice! More than halfway there!')
elif numsteps >= D_STEPS:
   print('Good start! Keep active the rest of the day!')
# Fill in the missing code here
   print('No worries. Still time to get some steps in!')
```

Correct Answer

```
# Constants for step levels
A_STEPS = 10000
B_STEPS = 8000
C_STEPS = 6000
D_STEPS = 4000

numsteps = int(input('How many steps did you walk today? '))

if numsteps >= A_STEPS:
   print('Great job!')
elif numsteps >= B_STEPS:
   print('Good work! Just a few more left to go.')
elif numsteps >= C_STEPS:
   print('Nice! More than halfway there!')
elif numsteps >= D_STEPS:
   print('Good start! Keep active the rest of the day!')
else:
   print('No worries. Still time to get some steps in!')
```

Complete Code Solution

```
# Constants for step levels
A_STEPS = 10000
B_STEPS = 8000
C_STEPS = 6000
D_STEPS = 4000

numsteps = int(input('How many steps did you walk today? '))

if numsteps >= A_STEPS:
   print('Great job!')
elif numsteps >= B_STEPS:
   print('Good work! Just 2000 left to go.')
elif numsteps >= C_STEPS:
   print('Nice! More than halfway there!')
elif numsteps >= D_STEPS:
   print('Good start! Keep active the rest of the day!')
else:
   print('No worries. Still time to get some steps in!')
```