```python
# Demonstrate a "while" loop.

# We commonly want to REPEAT some statements in our programs. A "loop" is a
# construct that lets us specify a section of code to repeat as long as some
# condition is true. In Python, one type of loop is a "while" loop.

# A few looping exercises:

# Loop until the user inputs the passcode
passcode = 12345
user_guess = int(input("Enter the passcode: "))
while user_guess != passcode:
    user_guess = int(input("Wrong, try again: "))

# What can we conclude if the program reaches this line?
print("That's the kind of passcode an idiot would have on his luggage!")


# Print the numbers from 1 to 100
number = 1
while number <= 100:
    print(number)
    number = number + 1


# Redo the student name example
student_number = 1
while student_number <= 5:
    student_name = input("Enter student {0}'s name: ".format(student_number))
    print("Welcome, {0}!".format(student_name))
    student_number = student_number + 1


# Greet a student by name.
user_input = 0
while user_input <= 0:
    user_input = int(input("Please enter a positive integer: "))

# Another validation example: enter a number from 1 to 10
user_input = 0
while not (user_input >= 1 and user_input <= 10):
    user_input = int(input("Please enter a number from 1 to 10: "))


# Lessons:
#   while loop
#   A few common loop patterns
#   "not" in conditions

```

```python
# Find the mean of a bunch of numbers
print("Please enter one number at a time. Enter a 0 to stop: ")

# We need variables to capture the user's input and sum it.
user_input = -1
user_sum = 0
total_numbers = 0

while user_input != 0:
    user_input = int(input())
    user_sum += user_input
    total_numbers += 1

# Now that we have the sum, we can find the average easily.
average = user_sum / total_numbers
print("The mean of those values is {0}".format(average))
# There is a BUG IN THIS CODE!!!




# Lessons:
#   Using an accumulator variable
#   += operator and its cousins -=, /=, *=, %=

```

```python
# Make a basic "menu" application that prints a menu with two options:
# 1. Say hello
# 2. Quit

# If the user selects 1, print "Hello" to the screen and loop again.
# If the user selects 2, quit the program.
# If the user selects anything else, loop again.
choice = 0
while choice != 2:
    print("1. Say hello")
    print("2. Quit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        print("Hello")
    elif choice == 2:
        print("Goodbye!")
    else:
        print("Follow the rules dummy!")


```

```python
import math

# Ask the user to input an integer, then determine if that integer is prime.
x = int(input("Enter an integer: "))

answer = True
divisor = 2
while divisor <= math.sqrt(x):
    if x % divisor == 0:
        answer = False
        break
    if divisor == 2:
        divisor = 3
    else:
        divisor += 2


if answer:
    print("It is prime!")
else:
    print("It is not prime!")

```

```
1  # Syntax vs. semantics errors.
2
3  # Syntax: errors in the structure of the program.
4  # 10a is not a valid name or literal
5  x = 10a
6
7  # strings must be terminated
8  x = "
9
10 # binary operators require two operands
11 x = 1 +
12
13 # parentheses must be closed
14 print("Hello!"
15
16 # conditions must be followed by a colon
17 if x > 10
18     pass
19
20 # blocks must be indented
21 if x == 100:
22 print("This is an error")
23
24
25 # Semantics: errors in the MEANING of the program
26 # using an undefined variable
27 x = y + 10
28
29 # using a value in a context that has no meaning
30 x = "hello" + 2
31
32 # using a non-Boolean expression as a condition
33 if x + 2:
34     pass
35
36 # passing incorrect types to functions
37 print(math.sqrt("Hello"))
38
39
40 # Lessons:
41 #   Syntax vs. semantic errors
42
```