Homework 6 is designed to give you a good preview of the format and content of the first midterm. It consists of 10 multiple choice questions and 3 programming problems.

When you take the midterm you will not have access to a Python programming environment. Therefore, it is essential to know how to read code and determine what it is going to do without running it. (This is very much like real life, where you should know what code will do even before you write it.)

For each multiple choice question, submit two answers: the one you get before you consult a Python interpreter and the one you get from the Python interpreter. When these two answers are different, include a sentence explaining why they are different.

Upload your answers as a .py file in Moodle.

Problem 1

```python
bool1 = 8%7 == 8//7
bool2 = 8/7 == 8//7
print(bool1 and bool2)
```

a. SyntaxError: invalid syntax
b. True False
c. True
d. False
e. none of the above

Problem 2

```python
billyGoatNum = 3
characters = ['littlestBillyGoat', 'middleBillyGoat', 'biggestBillyGoat', 'troll']
anObject = characters[billyGoatNum]
print(anObject)
```

a. 3
b. billyGoatNum
c. biggestBillyGoat
d. troll
e. none of the above
Problem 3

```python
spell = 'abbacadabra'
print(spell[4:7])
```

a. SyntaxError: invalid syntax
b. aca
c. acad
d. cada
e. none of the above

Problem 4

```python
states = ['NY', 'NJ', 'CT', 'CA', 'OR', 'WA', 'TX']
eastCoast = states[:3]
loneStar = states[-1]
print(eastCoast + loneStar)
```

a. IndexError: list index out of range
b. TypeError: can only concatenate list (not "str") to list
c. 'NY' 'NJ' 'CT' 'TX'
d. ['NY', 'NJ', 'CT', 'TX']
e. none of the above

Problem 5

```python
import turtle
swamp = turtle.Screen()
snapper = turtle.Turtle()
snapper.color('red')
snapper.fd(100)
```

a. a red line of length 100
b. turtle.TurtleGraphicsError: Unknown turtle type: 'snapper'
c. AttributeError: 'Turtle' object has no attribute 'fd'
d. invalid syntax: <string>
e. none of the above
Problem 6

def fullName(first, last):
    return first + ' ' + last
print(fullName('Mickey', 'Mouse'))

a. Mickey, , Mouse
b. Mickey Mouse
c. fullName: Mickey Mouse
d. NameError: name 'fullName' is not defined
e. none of the above

Problem 7

wishes = 'horses'
if wishes == 'horses':
    print('beggars ride')
if wishes == 'fishes':
    print('beggars eat')
else:
    print(wishes)

a. beggars ride
b. beggars ride
   horses
c. horses
d. " (the empty string)
e. none of the above

Problem 8

vowels = 'aeiou'
for letter in 'word':
    if letter not in vowels:
        print(letter)

a) NameError: 'word' is not defined
b) letter
   letter
   letter
c) w
   r
d
   d
d) nothing will be printed
e) none of the above
Problem 9

```python
for i in range(2):
    print(i)
```

a) i
b) i
i
c) 0
1
d) 1
2
e) none of the above

Problem 10

```python
def spell(word):
    for i in word:
        print(i)
spell('is')
```

a) ii
b) word word|
c) is
d) NameError: name 'spell' is not defined
e) none of the above
Programming problems 11-13 (20 pts each).

Problem 11

Part A: 10 points

Write a function named parallelLines(). The function parallelLines() takes four parameters:

1. a turtle
2. an integer, length, that is the length of each line
3. an integer, reps, that is the number of lines to draw
4. an integer, separation, that is the distance between parallel lines

For example, if length = 100, reps = 4 and separation = 10, the following is correct output

```
>_________
   _______
```

Part B: 10 points

Write code that calls parallelLines(), using the parameters in the example in part A. (Hint: before calling the function, you must import the turtle library and create a turtle.)

Problem 12

Write a function named outside() that tests whether a given number falls outside a specified range. The function outside() takes three parameters:

1. testNum
2. beginRange
3. endRange

The function outside() should return True if testNum is less than beginRange and greater than endRange. For example, the following would be correct output:

```
>>> print(outside(4, 1, 7))
False
```
Question 13

Write a function named greet(). The function greet() should ask the user’s name, and then greet the user by name, with a comma between the greeting and the name. The function greet() takes one parameter: a string named greeting.

For example, the following would be correct input and output:

```python
>>> greet('Cheerio')
What's your name? Gabriel
Cheerio, Gabriel
```