Programming Assignment 3
Due Friday, November 11 at 11:59 pm

This programming assignment has bugs walking on a number line of integers. Write the class `Bug` and assume that each bug created is moving on this number line. A bug's position is always expressed as an integer (which can be positive or negative). Your test code should exercise all parts of your code and be after

```
if __name__ == "__main__":
```

For the direction, either '+' or a word beginning with 'r' or 'R' will be interpreted as Right or facing toward the right end of the number line. A '-' or a word beginning with 'l' or 'L' will be interpreted as Left. Either case should be accepted and any other character will be ignored, that is, the direction will not be changed by setDir. In __init__ a bad direction value will raise ValueError.

The following methods are expected:

```
def __init__(self, initPos = 0, initDir = 'R'):
def move(self): # move one space in the current direction
def moveRandom(self): # move between 1 and 10 spaces in the current direction
def setPos(self, intPosition):
def getPos(self): # Return the current position (integer)
def setDir(self, dirStr): # See paragraph above
def getDir(self): # returns a direction string (either "right" or "left")
def __str__(self):
```

The positions of two bugs must be able to be compared so implement __eq__, __ne__, __lt__, __le__, __gt__, and __ge__. Obviously these each return a Boolean value. You should use only two instance variables (to indicate position and direction).

The string produced by __str__ should be like this: `Bug at -3 facing left`

Comment your code with your name on the first line. Imports should be done prior to any other Python code. Note that when testing moveRandom, answers will vary.

I will run your file to see what your test code does and also import the class (**which must be named Bug**) to exercise it with my test code. Submit to D2L as PA3.py. As usual, extra credit applies for being early.

Sample testing code and output that it should produce are on the next page.

```
if __name__ == "__main__":
    a = Bug()
    b = Bug(5)
    c = Bug(-2, "Left")
    d = Bug(0,'-')

    print("a:", a)
    print("b:", b)
    print("c:", c)
    print("d:", d)

    print("Testing comparisons: Bug a compared to b,
c, and d")
    bcd = [b,c,d]
    print("== test")
    for bug in bcd:
        print(a == bug, end = " ")
    print()

    print("!= test")
    for bug in bcd:
        print(a != bug, end = " ")
    print()

    print("< test")
    for bug in bcd:
        print(a < bug, end = " ")
    print()

    print("<= test")
    for bug in bcd:
        print(a <= bug, end = " ")
    print()

    print("> test")
    for bug in bcd:
        print(a > bug, end = " ")
    print()

    print(">= test")
    for bug in bcd:
        print(a >= bug, end = " ")
    print()

    a.move()
    d.move()
    b.moveRandom()
    c.moveRandom()
    print("a:", a)
    print("b:", b)
    print("c:", c)
    print("d:", d)

    try:
        e = Bug(10,"Wrong")
    except ValueError:
        print("Caught bad data in constructor.")

    a.setPos(100)
    a.setDir("leftward")
    print(a)
    print(a.getDir())
    print(a.getPos())
```

Output from this test code:

```
a: Bug at 0 facing right
b: Bug at 5 facing right
c: Bug at -2 facing left
d: Bug at 0 facing left
Testing comparisons: Bug a compared to b, c, and d
== test
False False True
!= test
True True False
< test
True False False
<= test
True False True
> test
False True False
>= test
False True True
a: Bug at 1 facing right
b: Bug at 6 facing right
c: Bug at -10 facing left
d: Bug at -1 facing left
Caught bad data in constructor.
Bug at 100 facing left
left
100
```