**Intro. to Computing (810:051)** **Lab 9** **Name:_____**

**Objectives**:

- Practice creating objects, calling methods, and passing objects as parameters
- Practice defining class definitions with inheritance

For today's lab you'll need several files, so download and extract the following file to your desktop:

http://www.cs.uni.edu/~fienup/cs051f10/labs/lab9.zip

This lab9 folder contains:

- a simple Die class (in the simple_die.py module) for a six-sided die,
- an AdvancedDie class (in the module advanced_die.py module) for a die which can be constructed with any number of sides. The AdvancedDie class inherits from the Die class, and
- testAdvancedDie.py file that creates some dice and calls their methods.

**Part A:** Run the testAdvancedDie.py script. After the script runs, the top-level varibles (e.g., die1, die2, and die3) remain loaded into the Python Shell environment. At the ">>>" prompt type the following:

a) `>>> die1.`

IDLE tells you the possible methods you can call for an AdvancedDie object like `die1`. What are they?

b) Now, continue typing a method name and a parathesis '(' as in:

```
>>> die1.getRoll(
```

IDLE provides you with a help message for the method. Where does this help message come from?

c) IDLE keeps track of the current type of each variable. You can use the `type( aVariable )` function to look up the type of `aVariable`. What is the result of each of the following?

```
>>> type(die1)

>>> type(die2)

>>> type(die3)
```

d) The `__add__` method of the AdvancedDie class allows you use the '+' operator with two AdvancedDie objects. What is the result of the following?

```
>>> die1 + die2
```

e) Explain the error message resulting from the following?

```
>>> die3 + die2
```

f) Try:

```
>>> die2 + die3
```

Why doesn't this give the same error message?

**After you have performed the above tasks, raise your hand and explain your results.**

**Part B: Implement a new subclass MoreAdvancedDie (in the new module more_advanced_die.py)** which inherits from the AdvancedDie class, and includes the following new features.

- An optional default keyword argument `label` to the `__init__` method which is stored in the string data attribute `_label`. If default value for this argument should be the empty string, i.e., "".
- An updated `__str__` method with includes the addition of the `_label` attribute
- A `getLabel` accessor method which returns the `_label` attribute string
- A `setLabel` mutator method that takes a string parameter which is used to update the `_label` attribute. We might invoke this method as:

```
myDie.setLabel("Lucky Die")   # sets myDie's label to "Lucky Die"
```

- For testing certain dice games, we might want a new method `setRoll` which takes as a parameter a roll value that is used to set a die's roll to a specified value. We might invoke this method as:

```
myDie.setRoll(3)   # sets myDie to a roll of 3
```

**After you have implemented and tested your MoreAdvancedDie, raise your hand and demonstrate your code.**

**Make sure that you log off the computer and take your USB drive before you leave.**