Do not open this exam until you are told. Read these instructions:

1. This is a closed book exam. No calculators, notes, or other aids are allowed. If you have a question during the exam, please raise your hand.
2. You must turn in your exam immediately when time is called at the end.
3. 6 problems which add up to 118 points total. 2 hours and 45 minutes. Each question’s point value is indicated.
4. In order to be eligible for as much partial credit as possible, show all of your work for each problem, write legibly, and clearly indicate your answers. Credit cannot be given for illegible answers.
5. After the last page there is paper for scratch work. If you need extra scratch paper after you have filled these areas up, please raise your hand. Scratch paper must be turned in with your exam, with your name and ID number written on it, but scratch paper will not be graded.
6. Please write out the following statement: “I pledge on my honor that I will not give or receive any unauthorized assistance on this examination.”

7. Fill in the following:

   NAME :

   SIGNATURE :

   G# :
1. Imagine you are working on a team to implement a database for storing information about types of animals. Answer the following questions:
   a. Write a class called Animal, stored in a file called animal.py, that has the following properties (31 points):
      i. Each animal has a color, weight, age, and species.
      ii. A constructor that takes as argument three of the attributes above (no more than three). It initializes those to the incoming values. The fourth argument should be initialized inside the constructor to whatever value you want.
      iii. **All of the fields must ALWAYS be of type string, for the lifetime of the object.**
      iv. A to-string method that can be called with a print statement, which would show the color and weight.
      v. A method that returns the age of the animal.
      vi. A method to set the age of the animal.
b. Now write a module, saved as driver.py, to create two Animal objects below (you can assign them any reasonable values). Set the age of the first animal to 15, and the age of the second animal to 1. There should be no errors when I run the module as python driver.py, assuming your Animal class and this file are in the same directory. (9 points)
2. Show all the output from running the code below (17 points):

```python
def func1(x,y,list):
    x = 5
    list.append(x)
    list.append(y)
    print "F1: " + str(list)
    list = []
    return list

def func2(x,y,list):
    print "F2: " + str(list)
    print "divide: " + str(x/y)
    print "mod: " + str(int(x)/int(y))

def func3(list = 2):
    print list
    return list

def main():
    list = [1]
    print list
    list.append([1])
    print func1(2,3,list)
    list = func1(4,5,[9])
    print list
    print func2(1.0,2,list)
    func2(3,4,list)
    print func3()
    list = func3([4,5])
    print list

main()
```
3. Write a function below called `find`, that calculates the number of times the character ‘c’ occurs in the elements of an incoming list (you may assume they will always be strings), **without using any built-in functions/methods**. The function should take a single list argument, and return an integer. Your function must not assume anything about the size of the incoming list, i.e., it could be empty. (15 points)

For example, `find(["cat", "cuckoo", "dog"])` would return 3

4. Write a function called `oddSum` that takes as argument a non-empty list of integers, and prints out the sum of every other integer in the list, **without using any built-in functions/methods**. (15 points)

For example, `oddSum([1, 4, 2, 4])` prints out 8
5. Give the output for the following code: (16 points)

```python
def func(x,y):
    print "div: " + str(int(x)/y)
y -= 1
    try:
        print "div2: " + str(x/y)
        print "done div"
        if x == 0:
            raise ValueError()
    except ZeroDivisionError:
        print "no div by 0!"
    print "done func"

def main(x,y):
    try:
        func(x,y)
        print "main"
    except ValueError:
        print "value error!"
    except Exception:
        print "other error!"
    finally:
        print "sigh"

main(2,3)
main("2",3)
main(0,3)
main(3,0)
```
6. Short answer (no more than three lines) (15 points)

   a. Write an expression to open a file called `best.html` for reading and writing (it’s in the current directory), and then write the string `hello` to the end file. Make sure your changes will persist on the hard drive.

   b. Write an expression to create an empty dictionary, and then insert the name “joe” with the value 11.

   c. Now write an expression to check if the dictionary above has the value “david” inside of it, and if so, prints out the value at that key.

   d. Imagine you are using the `Player` class from project 5, where the `updateHighScore` function increments the number of games played by one. What would the value of the `gamesPlayed` attribute be for the object pointed to by the variable `player` at the last line of the code below?

```
player = Player("Joe")
player.updateHighScore(11)
player.updateHighScore(0)
player = Player("Joe")
player.updateHighScore(13)
```
e. Give the output of the following code:
   ```python
   print (True or False)
   print (True and False)
   print (1 == 1 and 1 != 1)
   ```

f. What is the output of the following code?
   ```python
   list1 = [1,2,3]
   list2 = [list1, 2, 3]
   print list2
   list1.append(4)
   print list2
   print list1
   ```

g. Describe one thing you liked or found useful this semester, and one thing you did not like or did not find useful this semester.