Practice Problems for Exam 2

1. What will this print?

```
class A:
    def  __init__(self, x):
        self.value = x

    def  __str__(self):
        return str(self.value)
class B(A):
    def __init__(self, x):
        A.__init__(self, x)

def main():
    b = B(5)
    print(b)

main()
```

2. The folllowing program crashes. Why?

```
class C:
    def__init__(self, x):
        self.value = x

    def Print(self):
        print(x)

def main():
    c = C(23)
    c.Print()
main()
```

3. When I run the following program I get an error message that says method foo is being called with two arguments and it only has one. I only see one argument in the call foo(4) . What's up with this?

```
class D:
        def __init__(self, x):
                self.x = x

        def foo(y):
                return self.x+y

def main():
        d = D(32)
        print( d.foo(4) )

main()
```

4. Can objects of one class construct objects of another?
   Will the following program run correctly or give an error?

```
        class Wand:

                def __init__(self, name)
                        self.name = name

                def __str__(self):

                        return self.name

        class Wizard:

                        def __init___(self, name, wandName):

                                self.name = name
                                self.wand = Wand(wandName)

                        \def __str__(self):

                                return "Wizard %s has wand %s" %(self.name, self.wand)

        def main():
                        hp = Wizard("Harry Potter", "Elderwand")
                        print( hp )

        main()
```

5. Here is a program that uses a dictionary to store info about bands and albums, such as “Beatles” and “Revolver”. The dictionary is called Albums. The keys are names of bands; the value associated with each band is a list of the albums made by that band.

```
def main():
    Albums = { }

    done = True
    while not done:
        band = input( “Band name: “)
        if band == “”:
            done = True
        else:
            album = input( “Album name: “ )
            Albums[band].append(album)
```

This will crash the first time you try to add an album to the database. Why?

6. Here is a class definition. Write a main() function that creates Persons Harry and Luna, sets Harry’s age to 15, make Luna be Harry’s friend, and prints both of them.

```
class Person:
    def __init__(self, name):
        self.name = name
        self.age = 11
        self.friend = None  # a Person

    def setAge(self, a):
        self.age = a

    def setFriend(self, p):
        self.friend = p

    def __str__(self):
        if self.friend == None:
            return self.name
        else:
            return “%s whose friend is %s” %(self.name, self.friend.name)
```

7. Give code that creates a subclass HogwartsStudent of class Person from question (10). The only differences between a Person and a HogwartsStudent is that the latter have attributes year, which is set in the constructor to 1, and house, which can be any of the strings “Ravenclaw”, “Slitherin”, “Griffendor” and “Hufflepuff”. There should also be methods SetYear( ) and setHouse( ).

8. Create a class Library that holds a catalog of books. A Library has a list of books. A book has an author, a title, and a year. The Library class should have a method SearchByAuthor( ) that will print all of the books in the library by that author.

9. Write a program that asks the user over and over for the name of a person and one of that person’s favorite things. The program keeps track of all of this information and at the end prints all of the people and their list of favorite things. For example we might have the following interaction:

Person: Maria
Favorite: Raindrops on roses
Person: Maria
Favorite: Whiskers on kittens
Peron: bob
Favorite: python
Person: Maria
Favorite: Bright copper kettles
Person: Maria
Favorite: Warm woolen mittens
Person: <Return>

bob likes python
Maria likes Raindrops on roses Whiskers on kittens Bright copper kettles Warm woolen mittens

10. Here is function f.. What will f(23) print?

```
def f(x):
    if x == 0:
        print(“E”)
    elif x == 1:
        print(“O”)
    else:
        f(x-2)
```

11. What is wrong with this version of the factorial function? It keeps giving me a memory error.

```
def fact(n):
    return n*fact(n-1)
    if n == 1:
        return 1
```

12. Write a recursive substitute(a, b, s) that returns a string just like s only with every instance of variable b replaced by variable a. For example, substitute('s', 'b', "bob") returns "sos'

13. This one is a little harder than I'd probably put on an exam, but it is a nice challenge: Write a recursive function perm(n) that returns a list containing all of the permutations (orderings) of the numbers 1,2,3,…,n. For example, perm(1) returns [(1,)], perm(2) returns [(1,2), (2,1)] and perm(3) returns [(1,2,3), (1,3,2), (3,1,2), (2,1,3), (2,3,1), (3,2,1)].