

CSCI 150  
Exam 2 Solutions

1. Here are definitions of two classes. **Give a main() function that creates 3 persons: Hagrid, who is a Person, and Harry and Hermione, who are both Students at Hogwarts school. Make Harry and Hermione be friends.**

```
class Person:
    def __init__(self, name):
        self.name = name
        self.friends = [ ]

    def makeFriends(self, other):
        self.friends.append(other)
        other.friends.append(self)

    def __str__(self):
        s = self.name+" is friends with "
        for x in self.friends:
            s = s + x.name + " "
        return s

class Student(Person):
    def __init__(self, name, school):
        Person.__init__(self, name)
        self.school = school
```

Solution:

```
def main( ):
    p1 = Person( "Hagrid" )
    p2 = Student( "Harry", "Hogwarts" )
    p3 = Student( "Hermione", "Hogwarts" )
    p2.makeFriends(p3)
```

## 2. What will this program print?

```
class Top:
    def __init__(self, value):
        # value should be a string
        self.value = value

    def __str__(self):
        return self.value

class A(Top):
    def __init__(self, value):
        Top.__init__(self, value)

    def __str__(self):
        return self.value+self.value

class B(Top):
    def __init__(self,value):
        Top.__init__(self, value)

    def Print(self):
        print( "My value is " + self.value )

def main():
    t = Top("bob")
    a = A("bob")
    b = B("bob")
    print(t, a, b)

main()
```

Solution:

```
bob bobbob bob
```

3. Here is a program with a recursive function. **What will this program print?**

```
def blah(x):
    if x == 0:
        return 0
    elif x % 2 == 0:
        return 1 + blah(x//2)
    else:
        return blah(x-1)

def main():
    for x in range(0, 9):
        print(x, blah(x))

main()
```

Solution:

```
0 0
1 0
2 1
3 1
4 2
5 2
6 2
7 2
8 3
```

4. When I run the following program I get an error on the line  
print(f)

It says that `f` isn't defined. **Explain in English what is wrong with this program.**

```
class Foo:
    def __init__(self, f):
        self.value = f

    def Print(self):
        print(f)

def main():
    F = Foo(23)
    F.Print()

main()
```

Solution:

The variable `f` in the `Print` method is not an instance variable (its name doesn't start with `self`.) and it hasn't been given a value in the `Print` method, so it is undefined. The `Print` method can't see the variable `f` in the `__init__` method.

5. Here is another bad class definition:

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

    def addOne():
        self.x = self.x + 1

    def Print():
        print(self.x)

def main():
    b = Bar(13)
    b.addOne()
    b.Print()

main()
```

When I try to run this program I get an error message on the second line in main( ):  
b.addOne()

The error message says

TypeError: addOne() takes 0 positional arguments but 1 was given

**Explain in English what is wrong. What is the one argument in b.addOne() ??**

Solution:

The addOne method (and the Print method also) should have parameter (or argument) *self*. Since it doesn't, it takes no arguments. In the statement b.addOne( ), *b* is the argument that is passed to the function, and it should be given to the parameter *self*

6. **Write a recursive function `digits(x)` that returns the number of digits integer `x` has.** For example, `digits(9635)` returns 4. You can assume that `x` is never negative, so if `x` is less than 10 `digits(x)` should return 1; if `x` is between 10 and 99 it should return 2; if `x` is between 100 and 999 it should return 3, and so forth. Note that `len(str(x))` gives the right answer but that isn't what I am looking for. This should be a recursive function that works on the value of `x`.

Solution:

```
def digits(x):
    if x < 10:
        return 1
    else:
        return 1 + digits(x//10)
```

7. A game has 8 players; we'll call them "A", "B", "C", "D", "E", "F", "G", and "H". During the game they get points; we will model this with input statements. **Write a program with a loop that repeatedly asks for a player's name, then, if the name is not blank, ask for the number of points that player won. These points are added onto the player's total. When the game is over an empty string is entered for the player's name, and at that point the program prints the name of the player with the most points.** You don't need to worry about bad input; every player's name will be a letter in the range from "A" to "H", points will be non-negative integers. Here is a sample run:

```
player: B
points: 10
player: E
points: 12
player: H
points: 9
player: B
points: 3
player: <Return>
```

**player B wins!**

Solution:

```
def main():
    D = { }
    done = False
    while not done:
        player = input( "player: " )
        if player == "":
            done = True
        else:
            points = eval(input( "points: " ))
            if player in D.keys():
                D[player] = D[player] + points
            else:
                D[player] = points
    best = "A"
    for x in D.keys():
        if D[x] > D[best]:
            best = x
    print( "%s wins with %d points."%(best, D[best]))

main()
```

You can use this page as extra space for any question. If you want me to grade it be clear about what question you are answering here.

When you are finished write and sign the Honor Pledge.