https://goo.gl/WhSXRu

ITI1120 Review Session

Study Sheet

Inheritance and Recursion

Author: Faizaan Chishtie

Quick Review

Classes vs. Objects:

Object Is an instance of a: **Class**

What are classes?

Imagine you wanted to create a program that modified the position of a Tesla in 2D space:

Data:

x_coordinate
y_coordinate

Functions:

move_tesla()
will_crash() #checks if given Tesla will crash into a truck.

Let’s create a class named Tesla

Tesla

```
class Tesla:
    #dont use this
    area to declare
    variables
    def __init__(self, x,y):
        self.x_coordinate = x
        self.y_coordinate = y

    def move_tesla(self, x_move, y_move):
        “”” (number, number) -> None
            Changes Tesla’s x and y coordinate
        “””
        self.x_coordinate = x_move
        self.y_coordinate = y_move
```

This is the way we initialize classes.
Key words: “__init__” and “self”

This is a method! May change data inside an object.

A method is able to operate on data that is contained within the class (remembering that an object is an instance of a class - the class is the definition, the object is an instance of that data).

Inheritance

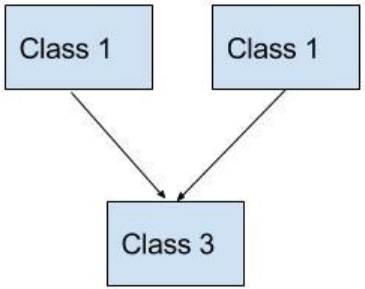

Fig: Mutiple Inheritance

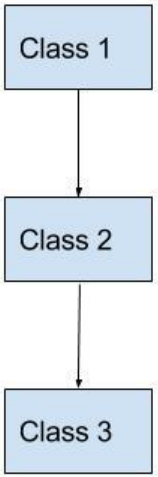

Fig: Multilevel Inheritance

Let’s make a class!

```
#code will be put here after session

class BaseClass:

  #Body of base class

class DerivedClass(BaseClass):

        BaseClass.__init__(#Constructor of base class)

        #in functions:

        → super(BaseClass, self).function()

        #Body of derived class
```

Recursion

Binary Search

```
def binary_search(alist, start, end, key):
    """Search key in alist[start... end - 1]."""
    if not start < end:
        return -1

    mid = (start + end)//2
    if alist[mid] < key:
        return binary_search(alist, mid + 1, end, key)
    elif alist[mid] > key:
        return binary_search(alist, start, mid, key)
    else:
        return mid
```

Tracing a binary search program

Fibonacci sequence:

```
def fibonacci(n):
    if(n <= 1):
        return n
    else:
        return(fibonacci(n-1) + fibonacci(n-2))
```