CS 120: Intro to Computer Programming II

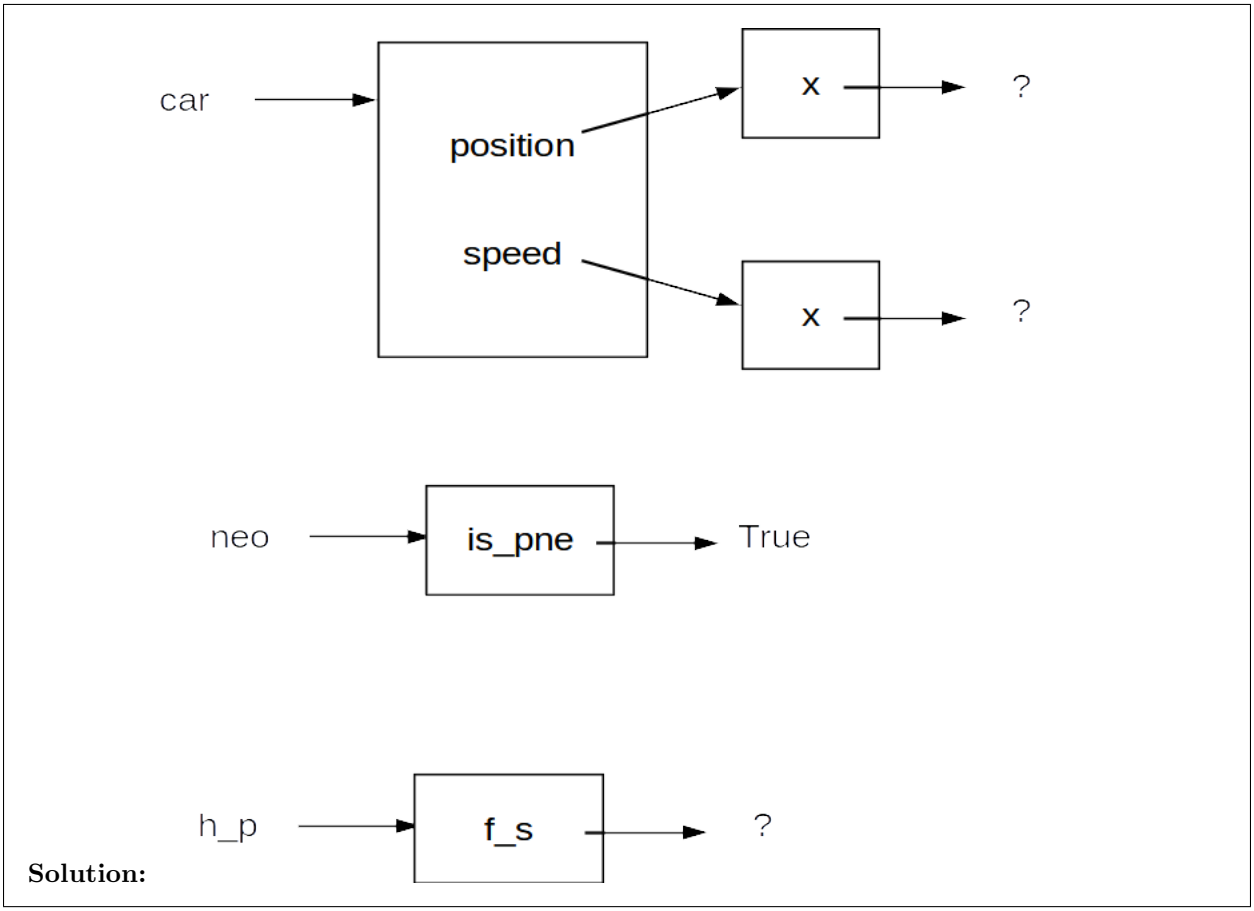
In-Class Activity - 04 Classes and Objects - Day 1

Activity 1 - Turn in this one

Examine the following code (yes, you did this in the video already), and draw a reference diagram for all of the objects that are mentioned in this snippet. You don't know all of the fields inside the objects, but draw as many as you can.

```
car.position.x += car.speed.x
neo.is_one = True
print(harry_potter.favorite_subject)
```

Solution:



Activity 2 - Turn in this one

Consider the following class. It has an init function, which simply sets a bunch of fields to `None`:

```
class Corner:
    def __init__(self):
        self.up    = None
```

```
        self.down  = None
        self.left  = None
        self.right = None
```

Now, suppose that we have built four of these objects, and linked them together like this:

```
nw = Corner()
ne = Corner()
se = Corner()
sw = Corner()
nw.right = ne
ne.down  = se
se.left  = sw
sw.up    = nw
```

As a group, copy this code into your favorite editor, and build the objects as I've stated here. But then write some code which will (a) print out the ID of all four objects, and also (b) print out the ID of each of the 4 fields, for each object.

It will be easier to write this code if you write a function, which takes one `Corner` as a parameter, and prints out everything about it. I'd suggest something along these lines:

```
def print_one(obj):
    print(id(obj))
    if obj.up is not None:
        ...
    ...
```

Solution:

```
def print_one(obj):
    print(f"obj: {id(obj)}")

    if obj.up is not None:
        print(f"  obj.up:    {id(obj.up)}")
    if obj.down is not None:
        print(f"  obj.down:  {id(obj.down)}")
    if obj.left is not None:
        print(f"  obj.left:  {id(obj.left)}")
    if obj.right is not None:
        print(f"  obj.right: {id(obj.right)}")

print_one(nw)
print_one(ne)
print_one(se)
print_one(sw)
```

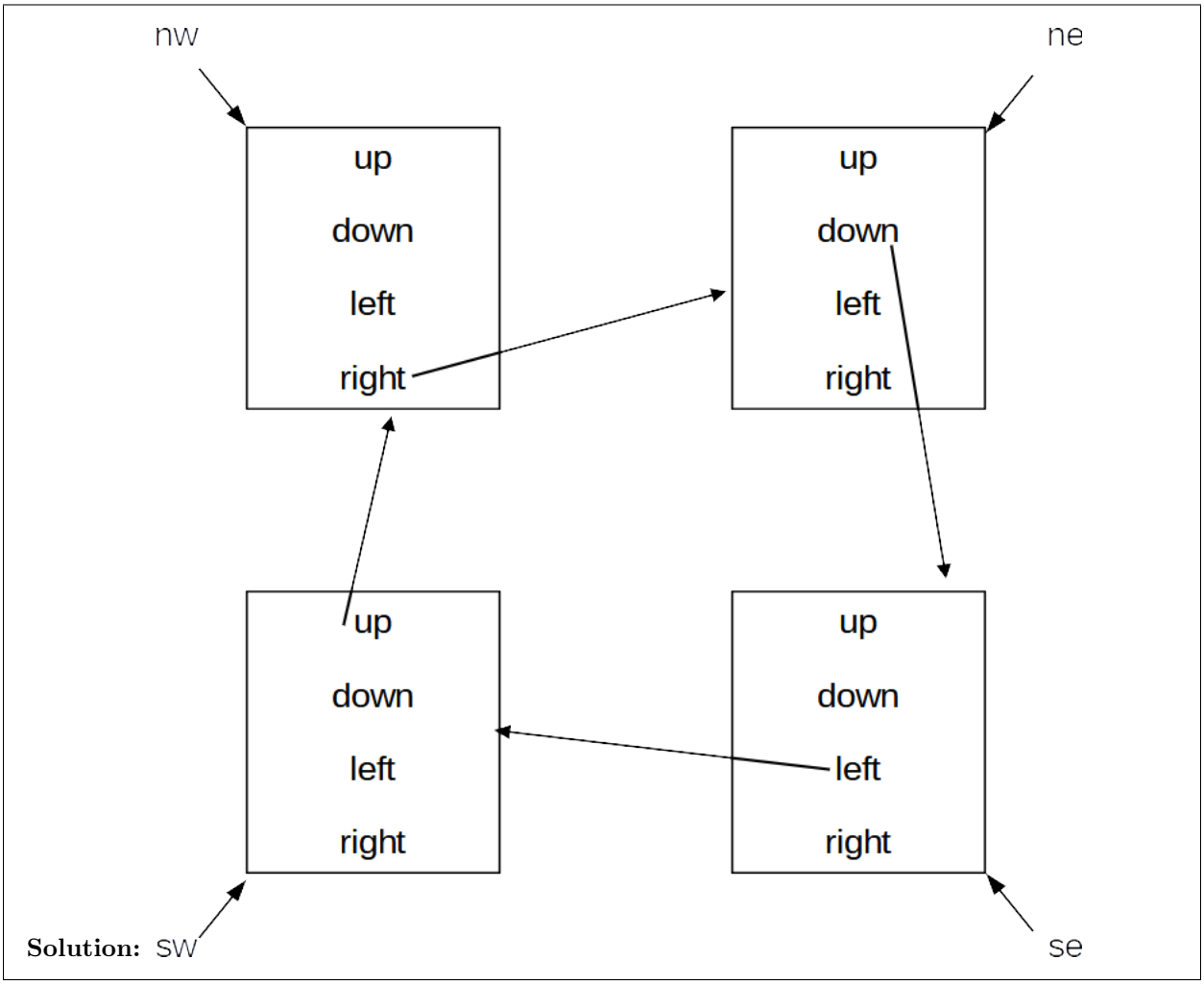
(activity continues on the next page)

Activity 3 - Optional

OPTIONAL. Complete this if you have time, and turn it in. If you don't have time, you may report to your TA that you ran out of time.

Using what you've learned from the IDs in the previous activity, draw a reference diagram of the four objects. Does it match what you would expect to see, from reading the code?

Solution:



Activity 4 - Optional

OPTIONAL. Complete this if you have time, and turn it in. If you don't have time, you may report to your TA that you ran out of time.

Finally, take the function that you wrote (you wrote one when I suggested it, right?) and move it into the class as a **method**. Your method must take one parameter (named `self`, which is an reference to the object itself.

Once you have the method ready, it should be possible to print everything out simply by typing:

```
nw.print_stuff()
ne.print_stuff()
```

```
se.print_stuff()
sw.print_stuff()
```

Solution:

```
class Corner:
    ...

    def print_one(self):
        print(f"self: {id(self)}")

        if self.up is not None:
            print(f"  self.up:    {id(self.up)}")
        if self.down is not None:
            print(f"  self.down:  {id(self.down)}")
        if self.left is not None:
            print(f"  self.left:  {id(self.left)}")
        if self.right is not None:
            print(f"  self.right: {id(self.right)}")

nw.print_one()
ne.print_one()
se.print_one()
sw.print_one()
```