Exam 2 – Practice Problems for the Paper-and-Pencil portion

1. Consider the code snippet below. It is a contrived example with poor style, but it will run without errors. What does it print when ***main*** runs?

```
def main():
    b = [44]
    a = (50, 30, 60, 77)
    x = 3

    for k in range(len(a)):
        b.append(a[x - k])

    print(a)
    print(b)
```

Write your answer in the box to the right.

Output:

```
(50, 30, 60, 77)

[44, 77, 60, 30, 50]
```

2. Consider the following two candidate function definitions:

```
def foo():
    print('hello')
```

```
def foo(x):
    print(x)
```

   a. Which is "better"? Circle the better function.

   b. Explain why you circled the one you did.

The second form allows the caller of the function to print ANYTHING, while the first is useful only for printing `'hello'`.

3. Short answer:

   a. What is the difference between a ***class*** and an ***instance of a class*** (in other words, the difference between a ***class*** and an ***object***)?

A *class* defines a *kind of thing*: What those things can do and what data they hold/know. An *object* (or *instance of a class*) is a *particular one* of that kind of thing, with its *particular values for its data*.

   b. Write a line or two of code that contains an example of each, clearly identifying the ***class*** and the ***object***.

point1 **and** ***point2*** **are** ***objects***

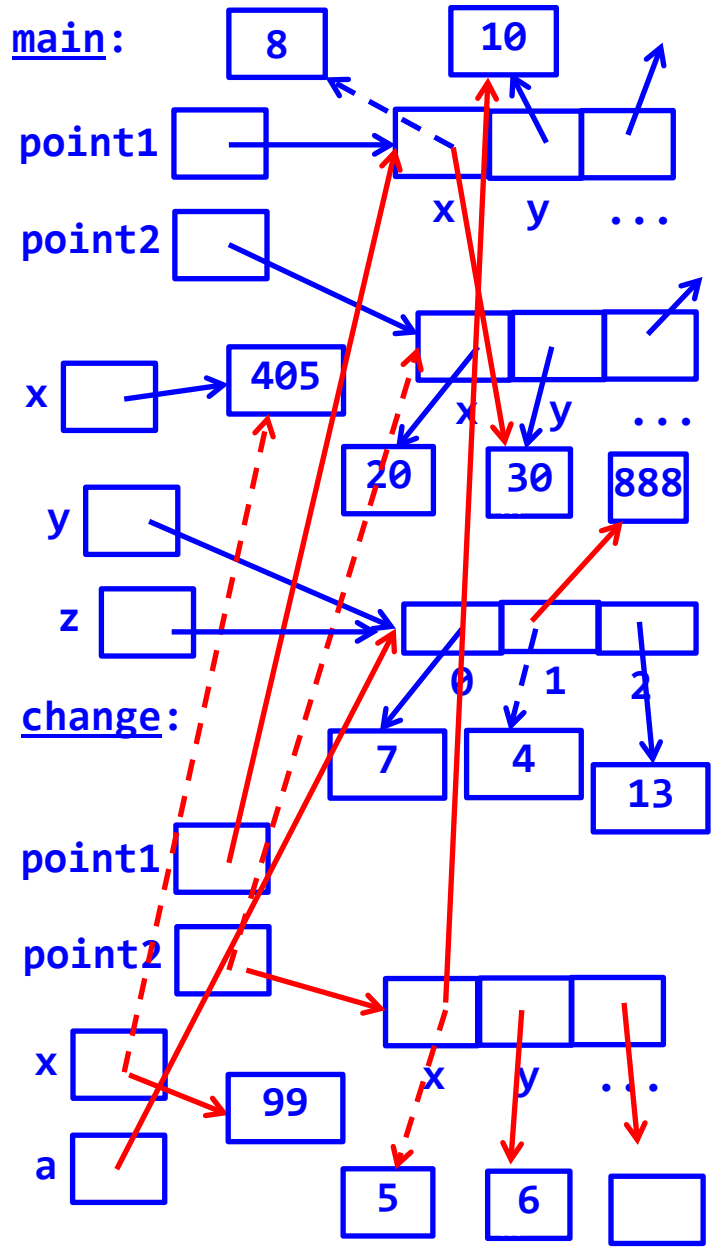
```
point1 = zg.Point(100, 44)
point2 = zg.Point(33, 1900)
```

They are both instances of the `zg.Point` *class*.

4. Consider the code in the below. To the right of the box of code, draw the ***box-and-pointer diagram*** for what happens when ***main*** runs. In the space below, show what the code would ***print*** when ***main*** runs.

```
import zellegraphics as zg

def main():
    point1 = zg.Point(8, 10)
    point2 = zg.Point(20, 30)
    x = 405
    y = [7, 4, 13]


    print('Before:',
          point1, point2, x, y)

    z = change(point1, point2, x, y)

    print('After:',
          point1, point2, x, y, z)


def change(point1, point2, x, a):
    point1.x = point2.y
    point2 = zg.Point(5, 6)
    point2.x = point1.y
    x = 99
    a[1] = 888
    print('Within:',
          point1, point2, x, a)

    return a
```

Draw box-and-pointer diagram below here



Dashed lines indicate arrows that are "Xed out".

What prints when *main* runs? (Assume that *points* get printed as per this example: **`Point(8, 10)`**.)

`Before:` **`Point(8, 10)`** **`Point(20, 30)`** **`405`** **`[7, 4, 13]`**

`Within:` **`Point(30, 10)`** **`Point(10, 6)`** **`99`** **`[7, 888, 13]`**

`After:` **`Point(30, 10)`** **`Point(20, 30)`** **`405`** **`[7, 888, 13]`** **`[7, 888, 13]`**