CSSE220 **Name:_____Section:____** Objects

Box&Pointer Part 2

Q1.

```
double pi = 3.14;
```

Q2.

```
int[] arr = new int[3];
arr[1] = 42;
```

Q3.

```
class Bar {
   public int value;
   public Bar() {
      value = 100;
   }
}
```

```
Bar b1 = new Bar();
Bar b2 = new Bar();
b2 = b1;
```

Q4.

```
class Foo {
   public int x;
   public Foo() {
      x = 0;
   }
}
```

```
Foo[] fooArr = new Foo[2];
fooArr[0] = new Foo();
fooArr[1] = fooArr[0];
fooArr[0].x = 20;
```

CSSE220 **Name:_____Section:____** Objects

Draw the following boxes and pointer diagrams. Use the following classes:

```
class Ninja {
   private String name;
   private int level;

   public Ninja(String name) {
       this.name = name;
       this.level = 1;
   }

   public void setLevel(int level) {
       this.level = level;
   }
}
```

```
class Pirate {
    private Ninja enemy;

    public Pirate(Ninja n) {
        this.enemy = n;
    }

    public Pirate() {
       this.enemy = new Ninja("Foo");
    }
}
```

Q5.

```
Pirate p = new Pirate(null);
```

Q6.

```
String name = "Buffalo";
Ninja[] ninjas = new Ninja[3];
for(int i = 0; i < 3; i++) {
      ninjas[i] = new Ninja(name);
      ninjas[i].setLevel(i);
}
```

Q7.

```
Ninja buffalo = new Ninja("Buffalo");
Ninja[] ninjas = new Ninja[3];
for(int i = 0; i < 3; i++) {
      ninjas[i] = buffalo;
      ninjas[i].setLevel(i);
}
```

Q8. What gets printed when you execute the following code:

```
HashMap<Integer, String> map1 = new HashMap<Integer, String>();
HashMap<Integer, String> map2 = new HashMap<Integer, String>();
map1.put(1, "One");
map2.put(1, "One");

System.out.println("map1 == map2: " + (map1 == map2));        output:_____

System.out.println("map1.equals(map2): " + map1.equals(map2));   output:_____
```

Q9. Use the following code to answer the next three questions:

```
public class StaticMystery {
	private String var;
	public StaticMystery(String var) {
		this.var = var;
	}
	public String getVar() {
		return this.var;
	}
	public void fun() {
		//cool code
	}
	public static void staticFun() {
		//more cool code
	}
	public static void main(String[] args) {

	}
}
```

Write code you would put in main to call fun() (hint: it's not static, so you'll need to construct an object first)

Write code you would put in main to call staticFun()

Why couldn't I change the function getVar() to be static?