1. What is your course section?
   Answer: _____

2. Because an array is an object, it has a `clone()`, `equals()`, and `toString()` methods.

   YES _____ NO _____

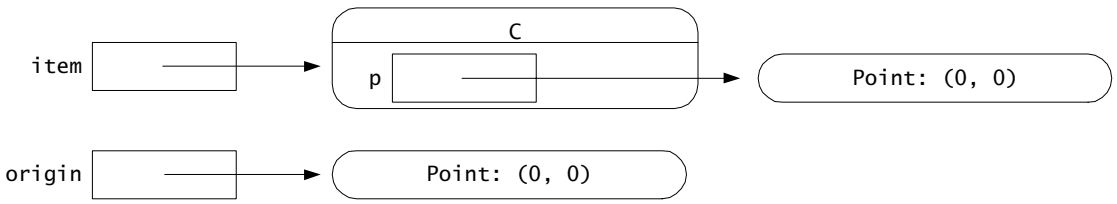
Questions 3 – 9 consider the following class `C`.

```
1.   public class C {
2.       private Point p;                // instance variable
3.
4.       public C(Point v) {             // specific constructor
5.           p = new Point(v.getX(), v.getY());
6.       }
7.
8.       public void set(Point v) {      // instance method
9.           p = v;
10.      }
11.
12.      public void perform(Point v) {  // instance method
13.          v.setX(10);
14.          p.setX(10);
15.      }
16.
17.      public void do(Point v) {       // instance method
18.          v = new Point(10, 0);
19.          p = v;
20.      }
21.      public static int f(int x) {    // class method
22.          return x + 7;
23.      }
24.
25.      public static void g(int x) {   // class method
26.          System.out.println( x + 7 );
27.      }
28.
29.  }
```

3. Consider the following code segment.

```
1.   Point origin = new Point(0, 0);
2.   C item = new C(origin);
```

   After it completes, memory has the following depiction.



   Do `origin` and `item`'s instance variable `p` (i.e., `item.p`) refer to *equivalent* `Point` objects after the following code segment?

   YES _____ NO _____

4. Consider the same code segment.

```
1.    Point origin = new Point(0, 0);
2.    C item = new C(origin);
```

Do `origin` and `item.p` refer to the *same* `Point` object after the preceding code segment finishes?

YES _____ NO _____

5. Consider the following code segment.

```
1.    Point origin = new Point(0, 0);
2.    C item = new C(origin);
3.    item.set(origin);
```

Do `origin` and `item.p` refer to the *same* `Point` object after the preceding code segment finishes?

YES _____ NO _____

6. Consider the following code segment.

```
1.    Point origin = new Point(0, 0);
2.    C item = new C(origin);
3.    item.perform(origin);
```

Do `origin` and `item.p` refer to the *same* `Point` object after the preceding code segment finishes?

YES _____ NO _____

7. Consider the following code segment.

```
1.    Point origin = new Point(0, 0);
2.    C item = new C(origin);
3.    item.do(origin);
```

Do `origin` and `item.p` refer to the *same* `Point` object after the preceding code segment finishes?

YES _____ NO _____

8. Consider the following code segment.

```
1.    int y = C.f(4);
```

Does it compile?

YES _____ NO _____

9. Consider the following code segment.

```
1.    int y = C.g(4);
```

Does it compile?

YES _____ NO _____

10. Write a *single statement* that defines and initializes an `int` array variable `unit` that is initialized to represent four 1s.

11. If you can, write a `static void` method `swap()` that takes two formal `int` parameters `x` and `y`. When invoked, the method is to interchange the values of its actual parameters. For example, with this method code segment

```
1.    int a = 10;
2.    int b = 11;
3.    swap(a, b);
4.    System.out.println(a + "   " + b);
```

should display

```
 11   10
```

If you cannot write the method explain why?

12. If you can, write a `static void` method `swap()` that takes three formal parameters: an `int` array `a` and two `int` variables `i` and `j`. When invoked, the method is to interchange the values of the `i`th and `j`th elements of the actual array parameter. For example, with this method code segment

```
1.    int[] list = new list[3];
2.    int m = 1;
3.    int n = 2;
4.    a[m] = 10;
5.    a[n] = 11;
6.    swap(list, m, n);
7.    System.out.println(list[m] + "   " + list[n]);
```

should display

```
11   10
```

If you cannot write the method explain why?

13. Suppose `b` is an *already* defined and initialized `int` array with 5 elements. Write a *single statement* that defines and initializes an `int` variable `i` to the value of the first element in `b`.

14. Suppose `b` is an *already* defined and initialized `int` array with 5 elements. Write a *single statement* that defines and initializes an `int` variable `i` to the value of the last element in `b`.

Questions 15 – 18 have you complete and use the following class `State`, where class `State` provides a representation of two values interest in regard to a state — its name and population size.

```
1.   public class State {
2.       private String name;   // represents name of the state
3.       private int size;      // represents number of people in the state
4.
5.       public State() {
6.           // TO BE FILLED IN **************
7.       }
8.
9.       public State(String s, int n) {
10.          name = s;
11.          size = n;
12.      }
13.
14.      public String toString() {
15.          return "(" + name + ": " + size + ")");
16.          p = v;
17.      }
18.
19.      public Object clone() {
20.          // TO BE FILLED IN **************
21.          return result;
22.      }
23.
24.      public boolean equals(Object v) {
25.          // TO BE FILLED IN **************
26.      }
27.  }
```

15. Write a *single statement* that can replace line 6 so that the default constructor initializes a `State` object to represent Virginia with its population of 7,078,515 people.

16. Write a *single statement* that can replace line 20 so that the `clone()` method produces a new object that is equivalent to the invoking (`this`) object.

17. Write a *code fragment* that can replace line 25 so that the `equals()` returns `true` if and only if `Object` parameter `v` is a `State` object equivalent to the invoking (`this`) object.

18. Write a *single statement* that defines a `State` array variable named `usa`. The definition should cause `usa` to reference a new `State` array with fifty elements.

19. Write a static `int[]` method `makeCopy()` that takes a single formal parameter `a`, where parameter `a` is an `int` array `a`. When invoked, the method returns a new `int` array of size `a.length`. The elements of this new array are equal to the corresponding elements of `a`.

20. Write a static `boolean` method `isSorted()` that takes a single formal parameter `a`, where parameter `a` is an `int` array `a`. When invoked, the method returns whether the elements of the list are in sorted order. Hint: an array `a` is sorted if and only if `a[i]` $\leq$ `a[i+1]` for all valid `i` and `i+1`.

PLEDGE