COMP 110/L
Fall 2022

Final Written Practice Exam with Solutions

1.) What is the output of the following code?

```
int x = 7;
if (x > 8) {
  System.out.println("Hello");
} else if (x < 10) {
  System.out.println("Something: " + x);
} else {
  System.out.println("Goodbye");
}
```

```
Something: 7
```

2.) What is wrong with the following code, if anything?

```
public static void foo(int x) {
  if (x == 8) {
    System.out.println(x);
  } else {
    System.out.println(42);
  }
  return 12;
}
```

Cannot return a value from a void method.

3.) Write a call to a method named `blah` with the parameters 7 and 8.

```
blah(7, 8)
```

4.) What is wrong with the following code, if anything?

```
public static int doSomething(int input) {
  switch (input) {
  case 1:
    return 2;
  case 2:
    return 3;
  }
}
```

```
Possible for this method not to return anything, even
though it's supposed to return an int.
For example, if the input is 4, no case will match.
```

5.) Consider the following method. What does `compute(1)` return?

```
public static int compute(int input) {
  int retval = 0;
  switch (input + 1) {
  case 0:
    retval++;
  case 1:
    retval += 2;
  case 2:
    retval += 3;
  case 3:
    retval += 6;
    break;
  case 4:
    retval += 12;
  }
  return retval;
}
```

9

6.) What is wrong with the following code, if anything?

```
public static void baz(int input) {
  if (input == 27) {
    return;
  } else {
    System.out.println(input);
  }
}
```

No problems (return without a value can be used for void methods).

7.) What is wrong with the following code, if anything?

```
public static int[] qwerty(int[] input) {
  return input[0];
}
```

qwerty is supposed to return an array of integers (int[]), but instead it returns an int (input[0] gets an int).

8.) Define a method named `add3` which takes three `int`s as parameters and returns their sum.

```
public static int add3(int x, int y, int z) {
  return x + y + z;
}
```

9.) What is the output of the following code?

```
int[] a = new int[]{ 5, 4, 3, 2 };
for (int x = 0; x < a.length - 1; x++) {
  System.out.println(a[x] + 1);
}
```

```
6
5
4
```

10.) What is the output of the following code?

```
int[] b = new int[]{ 9, 8, 7 };
for (int x = b.length - 1; x >= 0; x--) {
  System.out.println(b[x]);
}
```

```
7
8
9
```

11.) What is wrong with the following code, if anything?

```
int[] c = new int[0];
for (int x = 0; x <= c.length; x++) {
  System.out.println(c[x]);
}
```

```
Goes out of bounds.
The condition should be x < c
```

12.) What is the output of the following code?

```
int x = 3;
int y = 6;
int z = 10;
if (x > 2) {
  if (y < 8) {
    if (z > 5) {
      System.out.println(1);
    }
  } else {
    System.out.println(2);
  }
} else {
  System.out.println(3);
}
```

```
1
```

13.) What is the output of the following code?

```
int x = 0;
while (x < 5) {
  boolean b = false;
  if (x == 2) {
    b = true;
  }
  System.out.println(b);
  x++;
}
```

```
false
false
true
false
false
```

14.) What is the output of the following code?

```
int x = 0;
while (x < 5) {
  System.out.println("hi");
  x = 100;
  System.out.println("bye");
}
```

```
hi
bye
```

15.) Write a method named `randomOneThroughTen` that returns a random number between `1` and `10`, inclusive (i.e., both `1` and `10` are themselves possible numbers). You can assume `java.util.Random` has been imported. Do not use a seed value.

```
public static int randomOneThroughTen() {
  Random r = new Random();
  return r.nextInt(10) + 1;
}
```

16.) Write a method named `otherRandomOneThroughTen` that returns a random number between `1` and `10`, inclusive (i.e., both `1` and `10` are themselves possible numbers). `otherRandomOneThroughTen` takes a seed value as a parameter, and this seed value should be used as the initial value for random number generation. You can assume `java.util.Random` has been imported.

```
public static int otherRandomOneThroughTen(long seed) {
  Random r = new Random(seed);
  return r.nextInt(10) + 1;
}
```

17.) What is wrong with the following code, if anything?

```
public static void something(int x) {
  int x = 27;
  System.out.println("42");
}
```

`x is already declared as a parameter; both declarations of x are in the same scope.`

18.) What does the following code print, assuming it is run in jGrasp with jGrasp's “Run“ command?

```
public static void useX1(int x) {
  x = 5;
  System.out.println(x);
}
public static void useX2(int x) {
  useX1(x);
  System.out.println(x + 1);
  x = 8;
}
public static void main(String[] args) {
  int x = 42;
  useX2(x);
  System.out.println(x);
}
```

```
5
43
42
```

19.) List the types of the following expressions. If the expression has no type (it is ill-typed, so it'd produce a compile-time error), say so.

- `5 + 3`
- `5l + 3`
- `"42"`
- `"42" + 1`
- `5 + 5.2`
- `5.2 + 5l`
- `"hi" + true`
- `1 + true`

- `5 + 3`: `int`
- `5l + 3`: `long`
- `"42"`: `String`
- `"42" + 1`: `String`
- `5 + 5.2`: `double`
- `5.2 + 5l`: `double`
- `"hi" + true`: `String`
- `1 + true`: Error

20.) List the values produced by the following expressions.

- `5 - 2`
- `5 * 2`
- `5 / 2`
- `5 % 2`
- `6 - 2`
- `6 * 2`
- `6 / 2`
- `6 % 2`
- `5.0 - 2`
- `5.0 * 2`
- `5.0 / 2`
- `6.0 - 2`
- `6.0 * 2`
- `6.0 / 2`
- `5 - 2.0`
- `5 * 2.0`
- `5 / 2.0`
- `6 - 2.0`
- `6 * 2.0`
- `6 / 2.0`

- `5 - 2: 3`
- `5 * 2: 10`
- `5 / 2: 2`
- `5 % 2: 1`
- `6 - 2: 4`
- `6 * 2: 12`
- `6 / 2: 3`
- `6 % 2: 0`
- `5.0 - 2: 3.0`
- `5.0 * 2: 10.0`
- `5.0 / 2: 2.5`
- `6.0 - 2: 4.0`
- `6.0 * 2: 12.0`
- `6.0 / 2: 3.0`
- `5 - 2.0: 3.0`
- `5 * 2.0: 10.0`
- `5 / 2.0: 2.5`

- 6 - 2.0: 4.0
- 6 * 2.0: 12.0
- 6 / 2.0: 3.0

21.) What is wrong with the following code, if anything?

```
public static int p1(int x, int y) {
  return x + y;
}
public static void main(String[] args) {
  p1(5);
}
```

p1 takes two int arguments, but only one is provided in the call in main.

22.) What is wrong with the following code, if anything?

```
public static int p2(int x) {
  return x;
}
public static void main(String[] args) {
  p2(5, 6);
}
```

p2 takes one int argument, but two are provided in the call in main.

23.) What is wrong with the following code, if anything?

```
public static String p3(String x) {
  return x;
}
public static void main(String[] args) {
  p3(5);
}
```

p3 takes a String argument, but an int argument is passed in main.

24.) What is wrong with the following code, if anything?

```
public static String p4(int x) {
  return x;
}
```

p4 tries to return an int, but it's supposed to return a String based on its declaration.

25.) What is wrong with the following code, if anything?

```
public static int p5(int x) {
  return Integer.parseInt(x);
}
```

Integer.parseInt takes a String, but it's being passed an int

26.) Write a method named `firstPlusLast` that will take an array of integers and return the sum of the first and last integers. For example, if the first integer in the array is `3` and the last integer is `5`, then it should return `8`. If the array only contains one element (e.g., `7`), then the first and last integer in that array is the same (so for `7`, it should return `14`). If the array is empty, it should return `0`.

```
public static int firstPlusLast(int[] a) {
  if (a.length == 0) {
    return 0;
  } else {
    return a[0] + a[a.length - 1];
  }
}
```

27.) Convert the following `for` loop into a `while` loop.

```
for (int x = 0; x < 10; x++) {
  System.out.println(x);
}
```

```
int x = 0;
while (x < 10) {
  System.out.println(x);
  x++;
}
```

28.) Convert the following `while` loop into a `for` loop.

```
int x = 10;
while (x > 7) {
  System.out.println(x);
  x--;
}
```

```
for(int x = 10; x > 7; x--) {
  System.out.println(x);
}
```

29.) Convert the following `for` loop into a `while` loop.

```
for (int x = 0; x < 10; x += 2) {
  System.out.println(x);
  x--;
}
```

```
int x = 0;
while (x < 10) {
  System.out.println(x);
  x--;
  x += 2;
}
```

30.) For each loop below, state the number of iterations each loop will perform. If the loop is an infinite loop, state “infinite”.

- `for (int x = 0; x < 5; x++) { ... }`
- `for (int x = 0; x < 6; x += 2) { ... }`
- `for (int x = 5; x >= 0; x--) { ... }`
- `for (int x = 3; x > 10; x--) { ... }`
- `for (int x = 3; x >= 0; x) { ... }`

- `for (int x = 0; x < 5; x++) { ... }`: 5
- `for (int x = 0; x < 6; x += 2) { ... }`: 3
- `for (int x = 5; x >= 0; x--) { ... }`: 6
- `for (int x = 3; x > 10; x--) { ... }`: 0
- `for (int x = 3; x >= 0; x) { ... }`: infinite

31.) Write a method named `isMultipleOfThree` that will return `true` if a given `int` value is a multiple of 3, else `false`.

```
public static boolean isMultipleOfThree(int input) {
  return input % 3 == 0;
}
```

32.) Consider the following code and test suite:

```
public static int doSomethingStrange(int input) {
  if (input == 7) {
    return input;
  } else if (input == 8 || input < 5) {
    return input + 1;
  } else if (input > 25 && input < 100) {
    return input + 2;
  } else {
    return input + 3;
  }
}
@Test
public void test1() {
  assertEquals(doSomethingStrange(7), 7);
}
@Test
public void test2() {
  assertEquals(doSomethingStrange(-1), 0);
}
@Test
public void test3() {
  assertEquals(doSomethingStrange(50), 52);
}
```

The test suite above misses certain behaviors in the code. Add the **minimum** number of tests needed to cover all behaviors, with one `assertEquals` call per test.

```
@Test
public void test4() {
  assertEquals(doSomethingStrange(8), 9);
}
@Test
public void test5() {
  assertEquals(doSomethingStrange(100), 103);
}
```

33.) What is wrong with the following code, if anything?

```
public static int testMe(int x) {
  return x + 5;
}
@Test
public void test1() {
  assertEquals(testMe(7), 12);
}
public void test2() {
  assertEquals(testMe(0), 5);
}
```

Second test is missing @Test annotation.

34.) What is wrong with the following code, if anything?

```
public class Foo {
  private int x;
  public Foo(int y) {
    int x = y;
  }
  public int getX() {
    return x;
  }
}
```

The code compiles and runs, but probably doesn't work as desired.
The line “int x = y;” in the constructor declares a new local variable named “x”, instead of assigning to the instance variable “x”.
As such, the instance variable “x” is never assigned, so it will implicitly get initialized to 0 by Java.

35.) What is wrong with the following code, if anything?

```
public class Class1 {
  private int x;
  public Class1(int y) {
    y = x;
  }
}
```

Intention seems to be x = y; this does not set the instance variable x.

36.) What is wrong with the following code, if anything?

```
public class Baz {
  private int x;
  public Baz(int y) {
    x = y;
  }
  public static int getX() {
    return x;
  }
}
```

getX() is declared static, so it does not have access to the non-static instance variable x.

37.) What is wrong with the following code, if anything?

```
public class Blah {
  public static int x;
  public Blah(int y) {
    x = y;
  }
  public int getX() {
    return x;
  }
}
```

Technically nothing; the code will compile since instance methods have access to static items.
However, it's strange; getX() above could have been declared static for instance, and the name getX() implies that this is a getter for an instance variable.

38.) What is wrong with the following code, if anything (assume classes `Foo` and `Bar` are defined in `Foo.java` and `Bar.java`, respectively)?

```
public class Foo {
  public Foo() {}
  private int doSomething() {
    return 0;
  }
}
public class Bar {
  public static void main(String[] args) {
    Foo f = new Foo();
    f.doSomething();
  }
}
```

The doSomething method is marked as private in Foo, so it cannot be accessed in Bar, as Bar is not the same class as Foo.

39.) What is the output of the `main` method of `Class4` below?

```
public class Class2 {
  public Class2() {}
  public void m() {
    System.out.println("foo");
  }
}

public class Class3 {
  public Class3() {}
  public void m() {
    System.out.println("bar");
  }
}

public class Class4 {
  public static void main(String[] args) {
    Class2 x = new Class2();
    Class3 y = new Class3();

    x.m();
    y.m();
  }
}
```

```
foo
bar
```

40.) What is the output of the `main` method of `Class5` below?

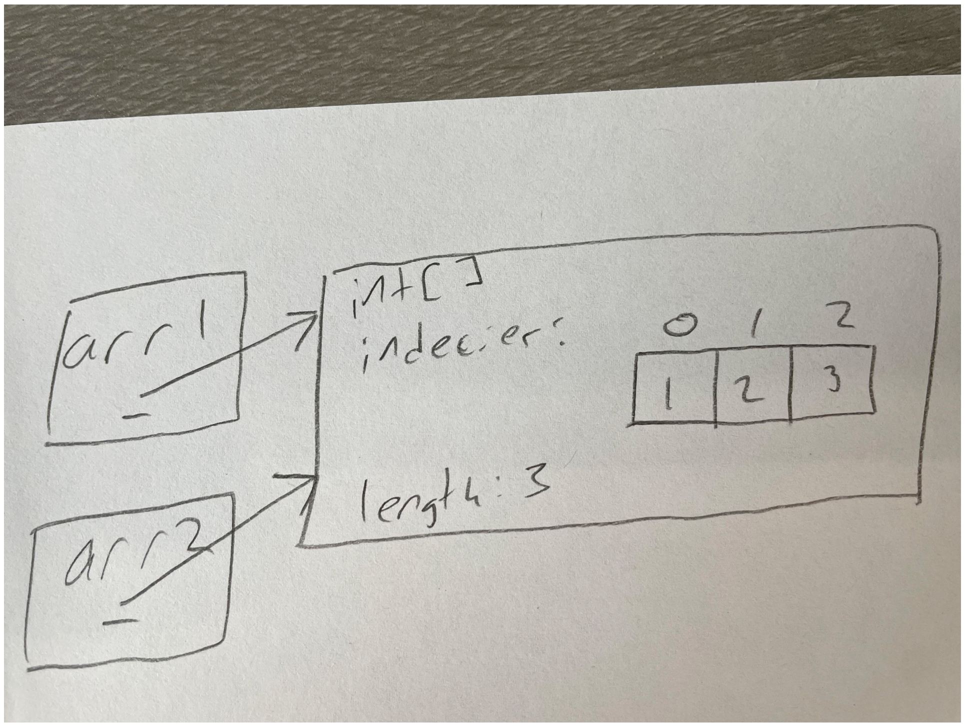
```
public class Class5 {
  private int x;
  public Class5(int y) {
    x = y;
  }
  public void foo(int x) {
    System.out.println(x);
  }
  public void bar(int y) {
    System.out.println(x);
  }
  public static void main(String[] args) {
    Class5 obj = new Class5(7);
    obj.foo(1);
    obj.bar(2);
  }
}
```

```
1
7
```

41.) Consider the following code snippet:

```
int[] arr1 = new int[]{1, 2, 3};
int[] arr2 = arr1;
```

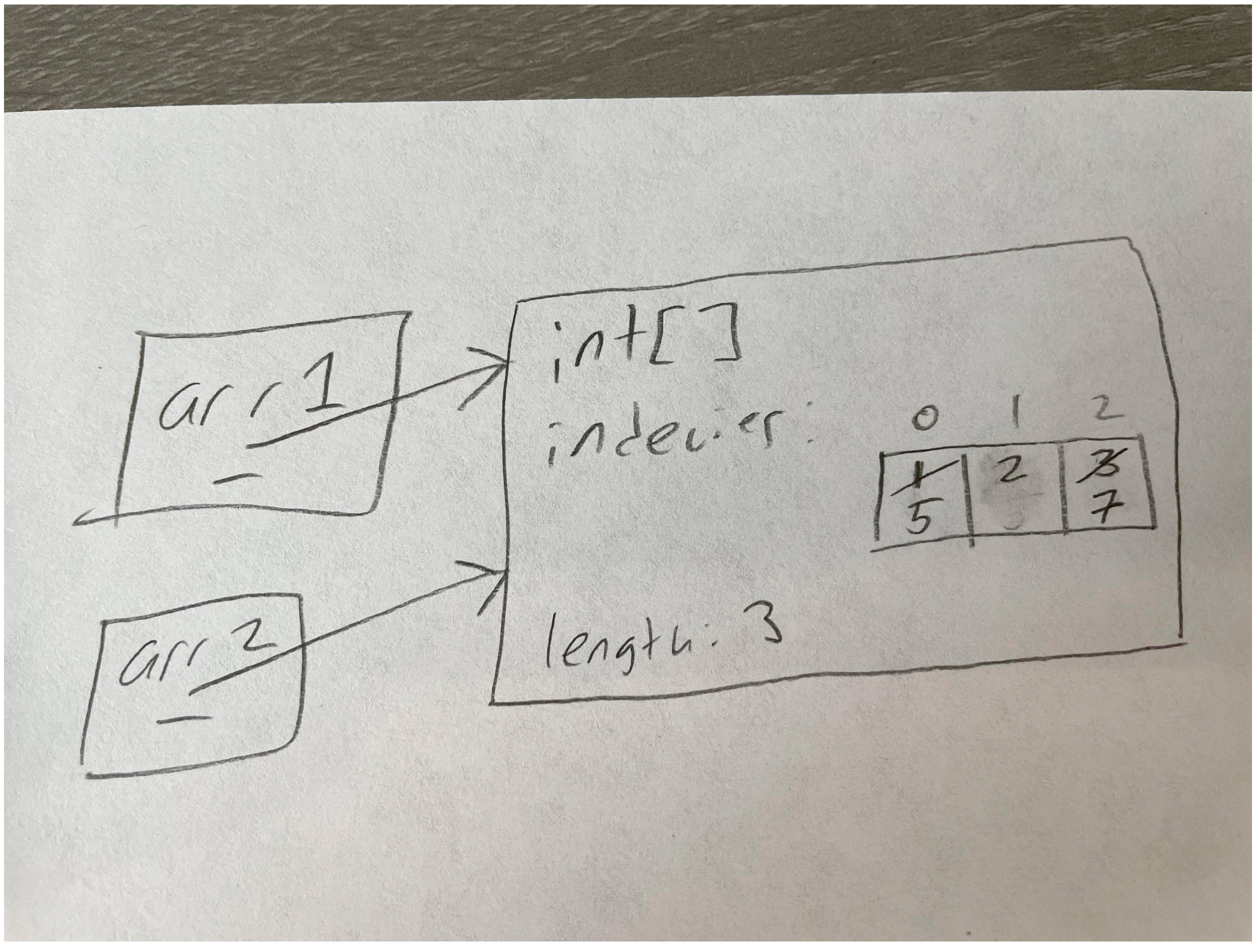
Write a memory diagram representing how memory looks after the above code snippet is executed.



42.) Consider the following code snippet:

```
int[] arr1 = new int[]{1, 2, 3};
int[] arr2 = arr1;
arr1[0] = 5;
arr2[2] = 7;
```

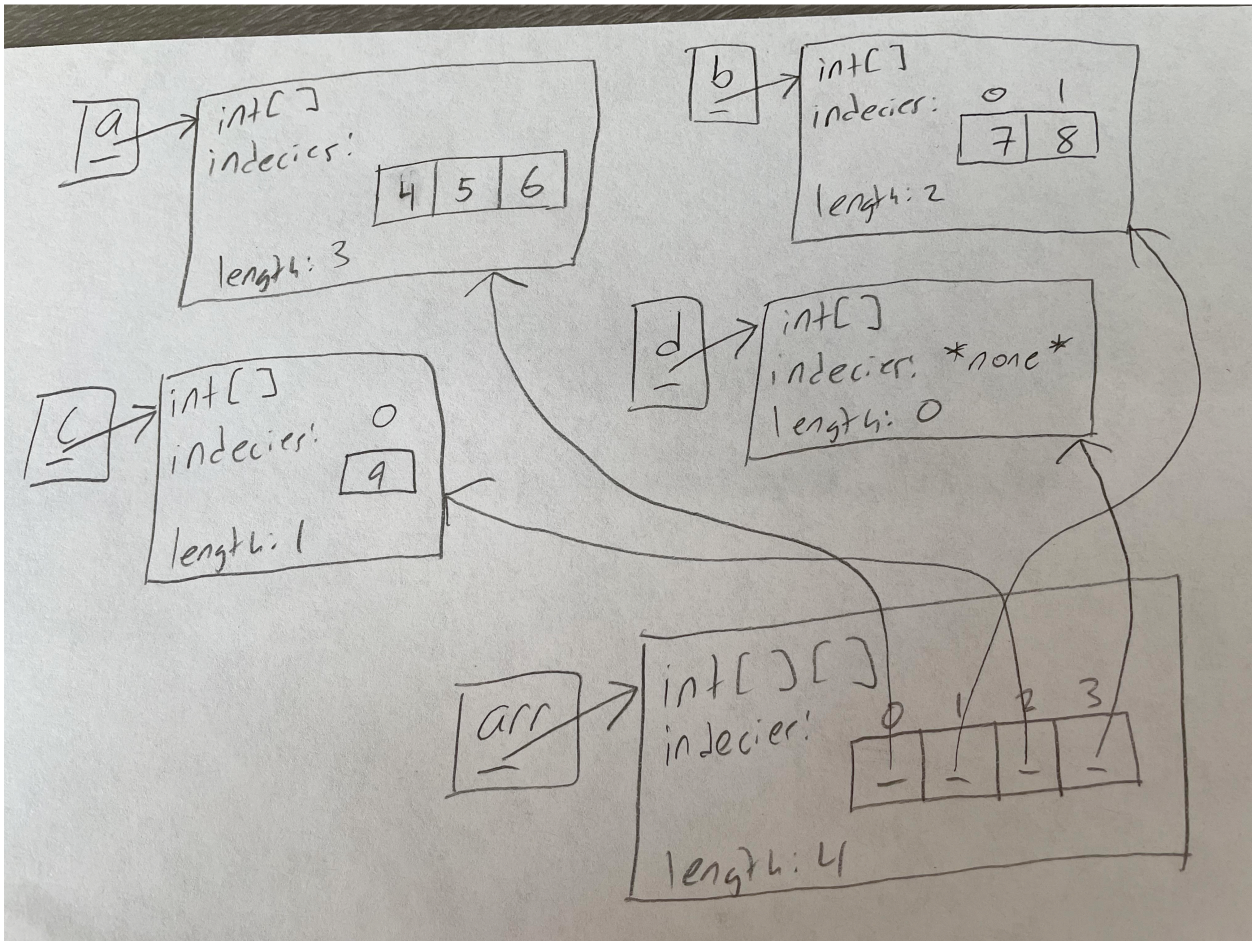
Write a memory diagram representing how memory looks after the above code snippet is executed.



43.) Consider the following code snippet:

```
int[] a = new int[]{4, 5, 6};
int[] b = new int[]{7, 8};
int[] c = new int[]{9};
int[] d = new int[0];
int[][] arr = new int[][]{a, b, c, d};
```

Write a memory diagram representing how memory looks after the above code snippet is executed.



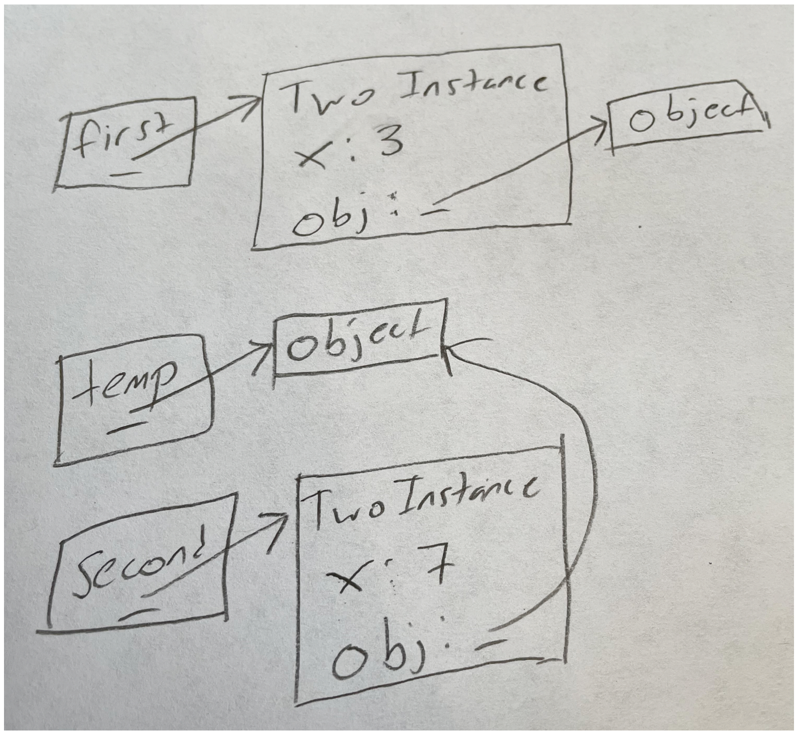
44.) Consider the following code snippet:

```
public class TwoInstance {
  private int x;
  private Object obj;

  public TwoInstance(int x, Object obj) {
    this.x = x;
    this.obj = obj;
  }

  public static void main(String[] args) {
    TwoInstance first = new TwoInstance(3, new Object());
    Object temp = new Object();
    TwoInstance second = new TwoInstance(7, temp);
    // HERE
    System.out.println();
  }
}
```

Assume we run the `main` method of the above program. Write a memory diagram representing how memory looks when `// HERE` is reached. You do **not** need to include `args` or `this` in your diagram.



45.) What is wrong with the following code, if anything?

```
public class Class1 {
  private int x;
  public Class1(int x) {
    this.x = x;
  }
}

public class Class2 extends Class1 {}
```

Class2 does not define a constructor that takes an int.

46.) What is wrong with the following code, if anything?

```
public class Class3 {
  protected int x;
  public Class3(int x) {
    this.x = x;
  }
}

public class Class4 extends Class3 {
  public Class4(int x) {
    super(x);
  }
  public int getX() {
    return x;
  }
}
```

No problems; Class4 now has access to x, since x is protected.

47.) What is wrong with the following code, if anything?

```
public class Class11 {}
public class Class12 extends Class11 {
  public static void m() {
    Class11 x = new Class12();
  }
}
```

```
No problems.  Class12 is-a Class11, so an instance of
Class12 can be assigned to a variable of type Class11.
```

48.) What is the output of the `main` method of `Class13` below?

```
public class Class14 {
  public void m() {
    System.out.println("foo");
  }
}

public class Class15 extends Class14 {
  public void m() {
    System.out.println("bar");
  }
}

public class Class13 {
  public static void main(String[] args) {
    Class14 x = new Class14();
    Class14 y = new Class15();
    Class15 z = new Class15();

    x.m();
    y.m();
    z.m();
  }
}
```

```
foo
bar
bar
```