CSC209 Data Structures (21 FEB 2018)
Practice Exercises

1. Inserting an item to the end of an unordered array
   a. takes time proportional to the size of the array
   b. requires multiple comparisons.
   c. Requires shifting other items to make room.
   d. takes the same time no matter how large the array is.
   (provided that the array is not full.)

2. O(1) means an algorithm operates in _________ time.

3. Either variables of primitive types or ______________ can be placed in an array.

4. Big O notation tells:

   a. The running time of an algorithm for a given size data structure.
   b. How the speed of an algorithm relates to the number of items.
   c. The running time of an algorithm for a given number of items.
   d. How the size of a data structure relates to the number of items.

5. Creating an array in Java requires using the keyword __________. For example, an expression used to create an array of TWELVE Strings is read __________________.

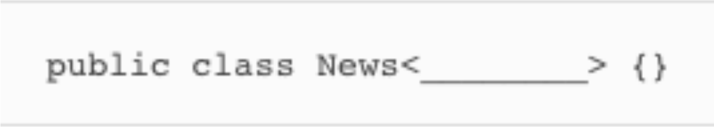
6. Which of the following is *not* true?

   a. A reference to a class object can be used to access public methods in the object.
   b. A reference to a class object has a size depending on its class.
   c. A reference to a class object has the data type of the class.
   d. A reference to a class object does not hold the object itself.

7. When you create a reference to a link in a linked list, it
   a. must refer to the first link.
   b. must refer to the link pointed to by **current**.
   c. must refer to the link pointed to by **next**.
   d. can refer to any link in the linked list.

8. How many references must you update in order to insert a link in the middle of a **singly linked list**?
   ____

9. Which of the following can fill in the blank to make the code compile (SELECT ALL THAT APPLY) ?

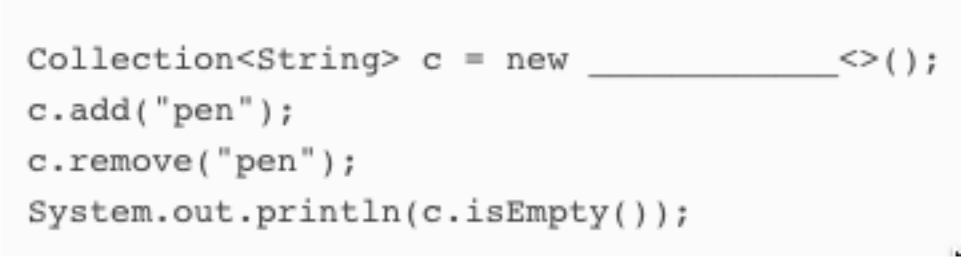


```
public class News<________> {}
```

a. ? b. News c. Object

10. Which method is available on both **List** and **Stream** implementations?
    a. filter()
    b. forEach()
    c. replace()
    d. sort()

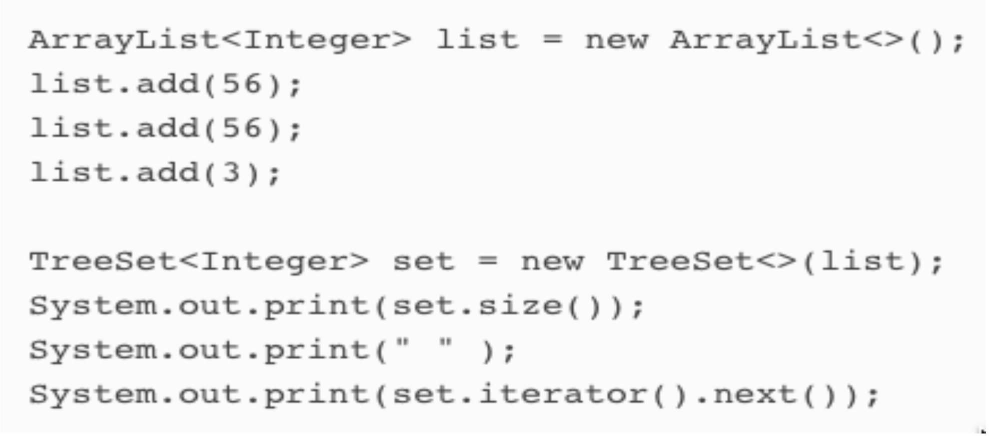
11. Which cannot fill in the blank for this code to compile?



```
Collection<String> c = new ____________<>();
c.add("pen");
c.remove("pen");
System.out.println(c.isEmpty());
```

    a. **ArrayDeque**
    b. **TreeMap**
    c. **TreeSet**
    d. All of these can fill in the blank.

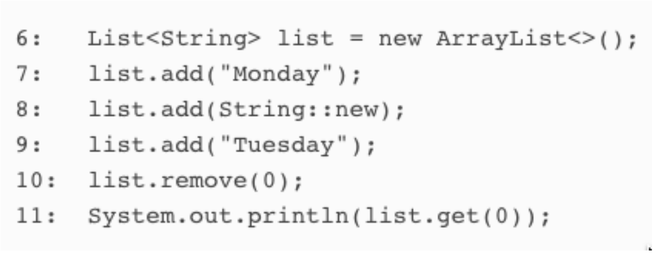
12. What is the result of the following?



```
ArrayList<Integer> list = new ArrayList<>();
list.add(56);
list.add(56);
list.add(3);

TreeSet<Integer> set = new TreeSet<>(list);
System.out.print(set.size());
System.out.print(" " );
System.out.print(set.iterator().next());
```

    a. 2 3
    b. 2 56
    c. 3 3
    d. 3 56

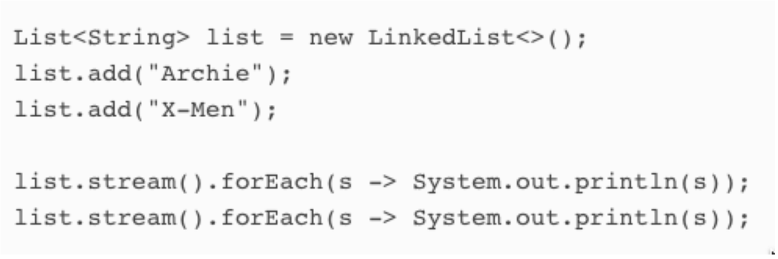
13. What is the result of the following?



```
List<String> list = new ArrayList<>();
list.add("Monday");
list.add(String::new);
list.add("Tuesday");
list.remove(0);
System.out.println(list.get(0));
```

a. An empty String
b. Monday
c. The code does not compile.
d. The code compiles but throws an exception at runtime.

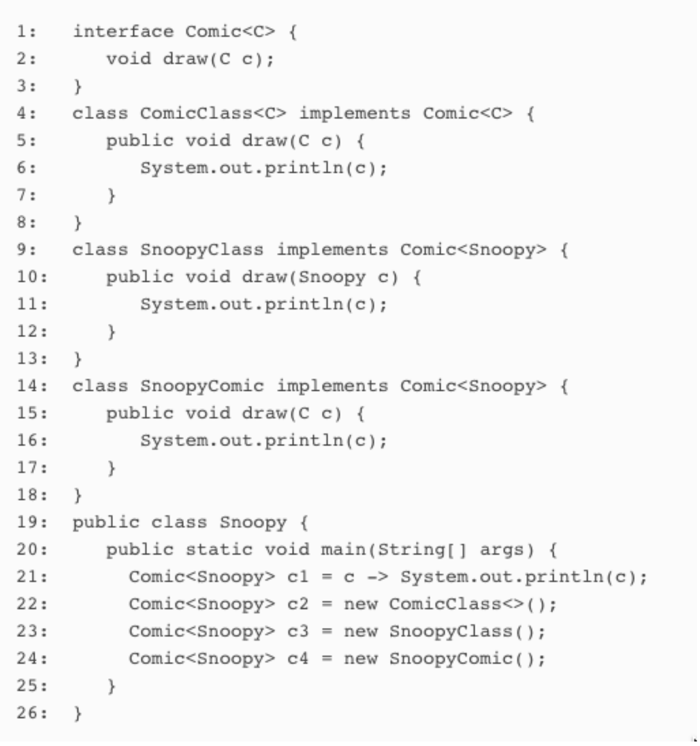
14. How many lines does this code output?



```
List<String> list = new LinkedList<>();
list.add("Archie");
list.add("X-Men");

list.stream().forEach(s -> System.out.println(s));
list.stream().forEach(s -> System.out.println(s));
```

a. Two
b. Four
c. The code does not compile.
d. The code compiles but throws an exception at runtime.

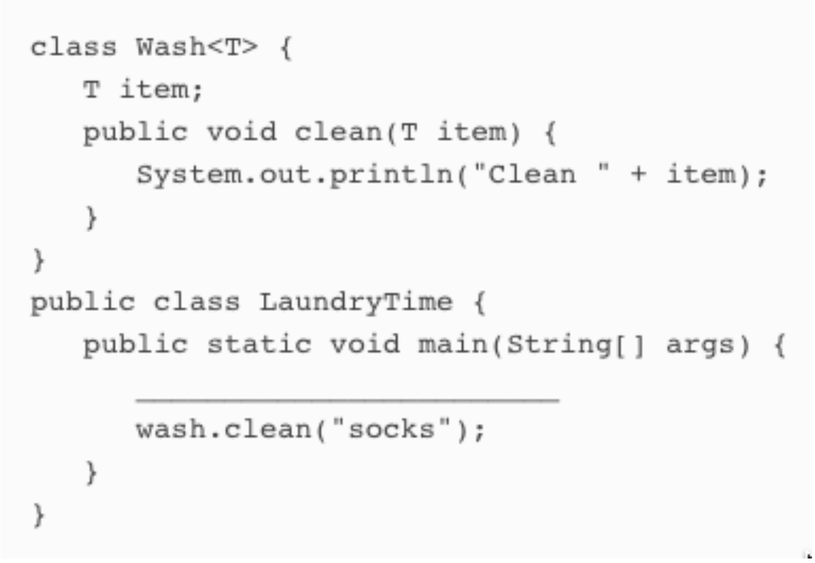
15. Which line in the **main()** method does *not* compile or points to a class that does *not* compile?



```
1:   interface Comic<C> {
2:      void draw(C c);
3:   }
4:   class ComicClass<C> implements Comic<C> {
5:      public void draw(C c) {
6:         System.out.println(c);
7:      }
8:   }
9:   class SnoopyClass implements Comic<Snoopy> {
10:     public void draw(Snoopy c) {
11:        System.out.println(c);
12:     }
13:  }
14:  class SnoopyComic implements Comic<Snoopy> {
15:     public void draw(C c) {
16:        System.out.println(c);
17:     }
18:  }
19:  public class Snoopy {
20:     public static void main(String[] args) {
21:       Comic<Snoopy> c1 = c -> System.out.println(c);
22:       Comic<Snoopy> c2 = new ComicClass<>();
23:       Comic<Snoopy> c3 = new SnoopyClass();
24:       Comic<Snoopy> c4 = new SnoopyComic();
25:     }
26:  }
```

a. Line 21
b. Line 22
c. Line 23
d. Line 24

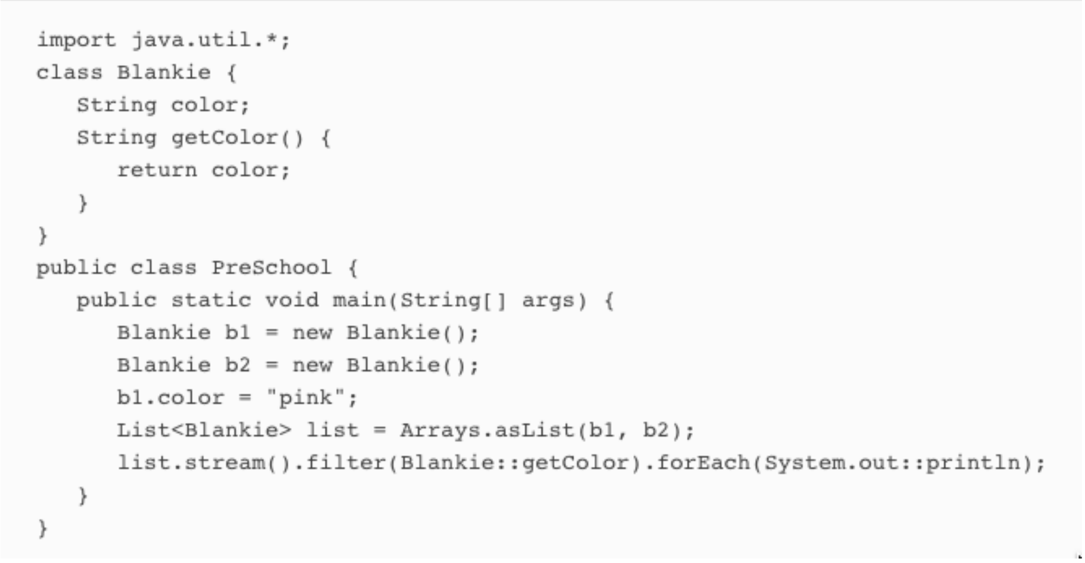
16. Which option cannot fill in the blank to print **Clean socks**?



```
class Wash<T> {
   T item;
   public void clean(T item) {
      System.out.println("Clean " + item);
   }
}
public class LaundryTime {
   public static void main(String[] args) {
      ________________________
      wash.clean("socks");
   }
}
```

a. Wash wash = new Wash();
b. Wash wash = new Wash<String>();
c. Wash<String> wash = new Wash<>();
d. All three can fill in the blank.

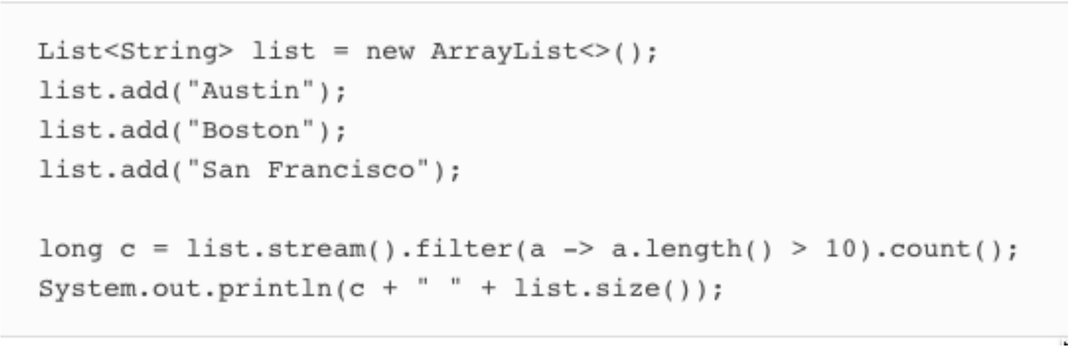
17. How many lines does the following code output?



```
import java.util.*;
class Blankie {
   String color;
   String getColor() {
      return color;
   }
}
public class PreSchool {
   public static void main(String[] args) {
      Blankie b1 = new Blankie();
      Blankie b2 = new Blankie();
      b1.color = "pink";
      List<Blankie> list = Arrays.asList(b1, b2);
      list.stream().filter(Blankie::getColor).forEach(System.out::println);
   }
}
```

a. One
b. Two
c. The code does not compile.
d. The code compiles but throws an exception at runtime.

18. What does the following output?



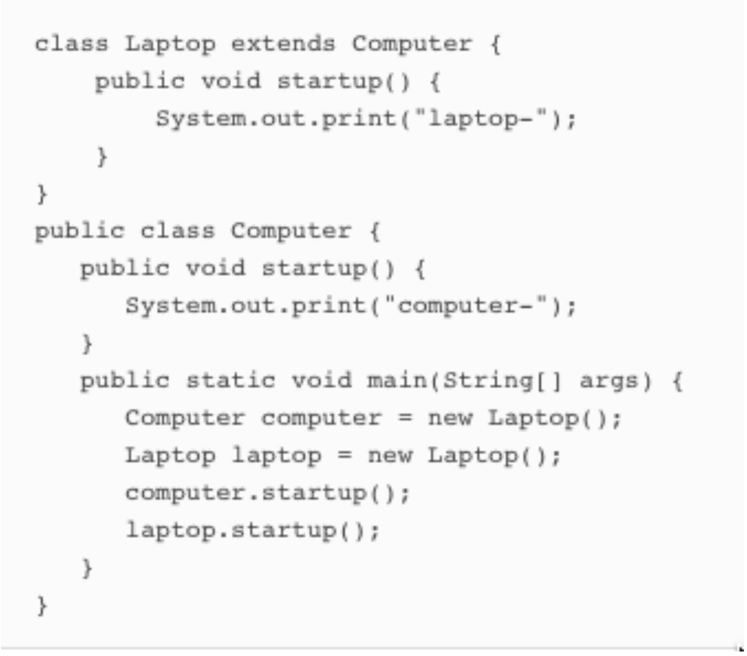
```
List<String> list = new ArrayList<>();
list.add("Austin");
list.add("Boston");
list.add("San Francisco");

long c = list.stream().filter(a -> a.length() > 10).count();
System.out.println(c + " " + list.size());
```

a. 1 1
b. 1 3
c. 2 3
d. None of the above

19. Which answer choice can replace **LINE 6** so the code continues to produce the same output?

a. System.out.println(rug.asString);
b. System.out.println(rug.asString());
c. System.out.println(rug.toString);
d. System.out.println(rug.toString());

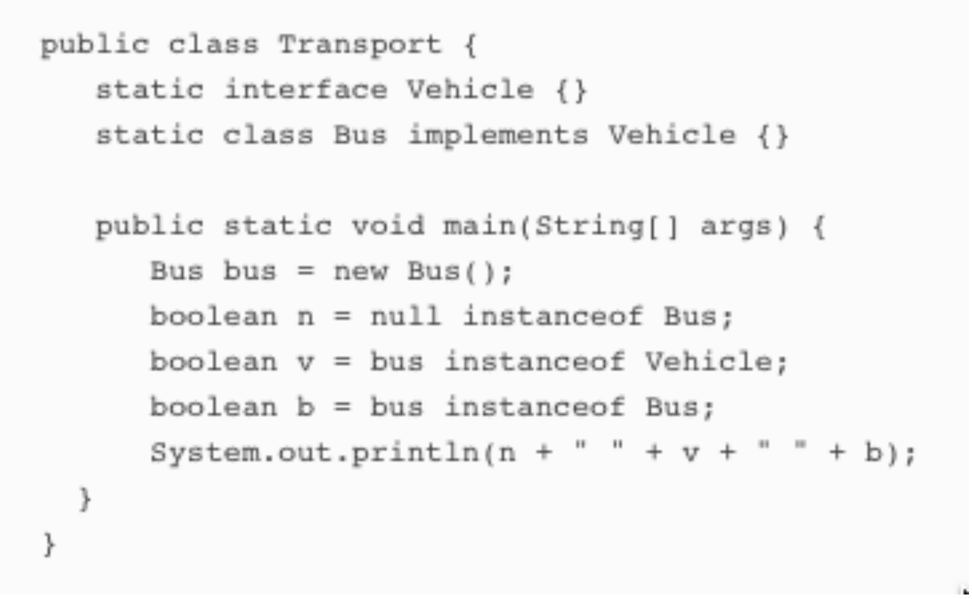
20. What is the output of this code?



```
class Laptop extends Computer {
    public void startup() {
        System.out.print("laptop-");
    }
}
public class Computer {
   public void startup() {
      System.out.print("computer-");
   }
   public static void main(String[] args) {
      Computer computer = new Laptop();
      Laptop laptop = new Laptop();
      computer.startup();
      laptop.startup();
   }
}
```

a. computer-laptop-
b. laptop-computer-
c. laptop-laptop-
d. None of the above

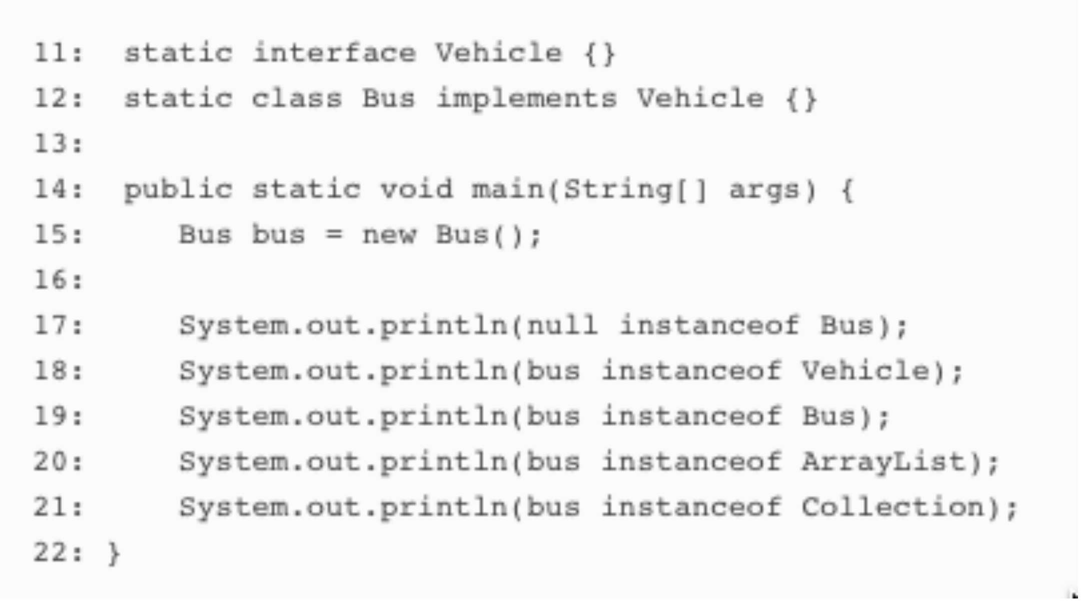
21. What does the following print?



```
public class Transport {
   static interface Vehicle {}
   static class Bus implements Vehicle {}

   public static void main(String[] args) {
      Bus bus = new Bus();
      boolean n = null instanceof Bus;
      boolean v = bus instanceof Vehicle;
      boolean b = bus instanceof Bus;
      System.out.println(n + " " + v + " " + b);
  }
}
```

a. true true true
b. false true true
c. false false false
d. None of the above

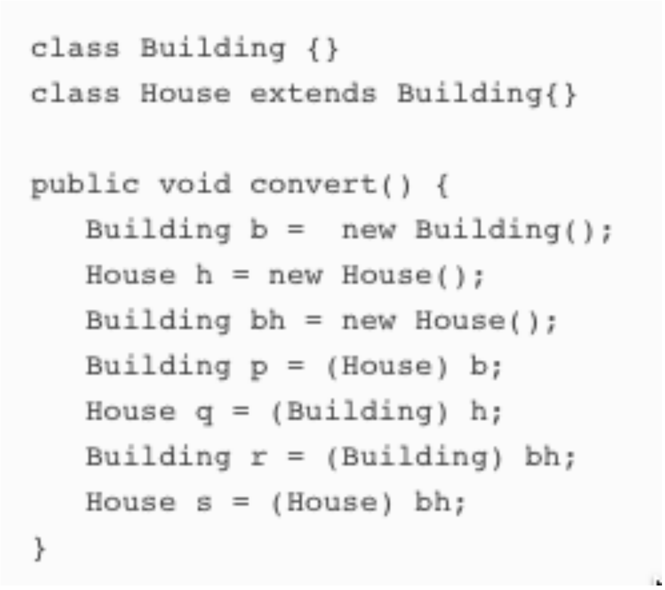
22. How many lines of the main method fail to compile?



```
static interface Vehicle {}
static class Bus implements Vehicle {}

public static void main(String[] args) {
   Bus bus = new Bus();

   System.out.println(null instanceof Bus);
   System.out.println(bus instanceof Vehicle);
   System.out.println(bus instanceof Bus);
   System.out.println(bus instanceof ArrayList);
   System.out.println(bus instanceof Collection);
}
```

a. One
b. Two
c. Three
d. Four

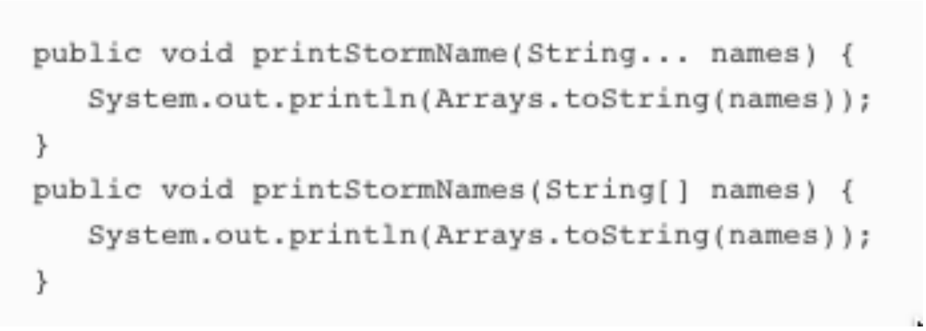
23. Which variable declaration is the first line not to compile?



```
class Building {}
class House extends Building{}

public void convert() {
   Building b =  new Building();
   House h = new House();
   Building bh = new House();
   Building p = (House) b;
   House q = (Building) h;
   Building r = (Building) bh;
   House s = (House) bh;
}
```

a. p
b. q
c. r
d. s

24. Given the following two methods, which method call will not compile?



```
public void printStormName(String... names) {
   System.out.println(Arrays.toString(names));
}
public void printStormNames(String[] names) {
   System.out.println(Arrays.toString(names));
}
```

a. printStormName(“Arlene”);
b. printStormName(new String[] {“Bret”});
c. printStormName(“Cindy”);
d. printStormName(new String[] { “Don” });

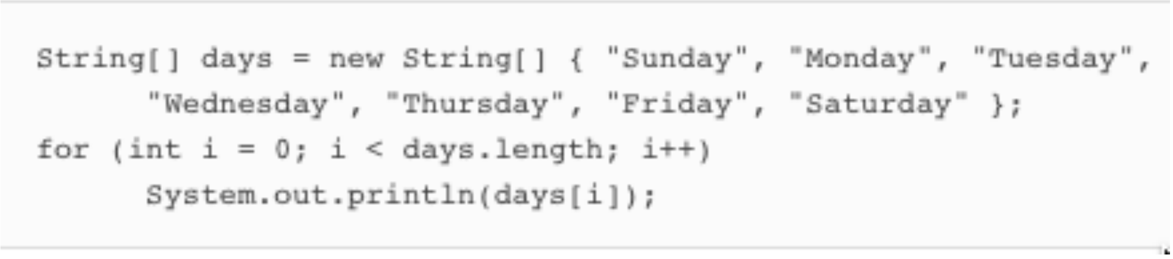
25. How do you determine the number of elements in an array?

a. array.length
b. array.lengt()
c. array.size
d. array.size()

26. Which of the following create an empty two-dimensional array with dimensions 2x2 ?

a. int[][] blue = new int[2,2];
b. int[][] blue = new int[2], [2];
c. int[][] blue = new int[2][2];
d. int[][] blue = new int[2 x 2];

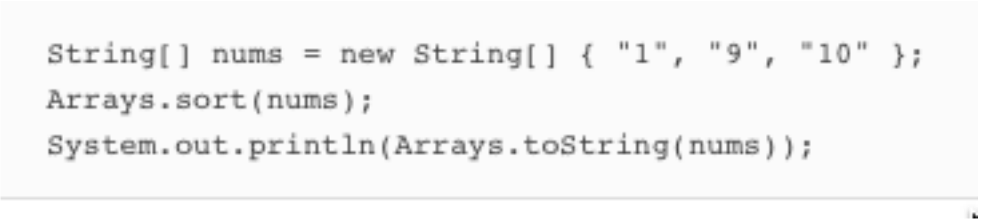
27. How many lines does the following code output?



```
String[] days = new String[] { "Sunday", "Monday", "Tuesday",
      "Wednesday", "Thursday", "Friday", "Saturday" };
for (int i = 0; i < days.length; i++)
      System.out.println(days[i]);
```

a. Six
b. Seven
c. The code does not compile.
d. The code compiles but throws an exception at runtime.

28. What does this code output?

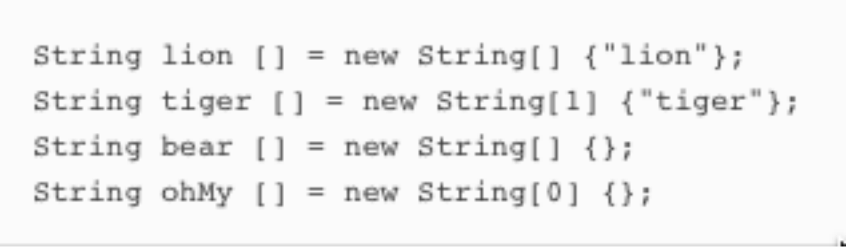


```
String[] nums = new String[] { "1", "9", "10" };
Arrays.sort(nums);
System.out.println(Arrays.toString(nums));
```

a. [1, 9, 10]
b. [1, 10, 9]
c. [10, 1, 9]
d. None of the above

29. Which of the following references the first and last element in a non-empty array?
a. trains[0] and trains[trains.length]
b. trains[0] and trains[trains.length – 1]
c. trains[1] and trains[trains.length]
d. trains[1] and trains[trains.length – 1]

30. How many of the following are legal declarations?



```
String lion [] = new String[] {"lion"};
String tiger [] = new String[1] {"tiger"};
String bear [] = new String[] {};
String ohMy [] = new String[0] {};
```

a. None
b. One
c. Two
d. Three

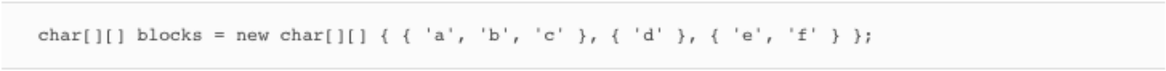
31. Which is not a true statement about an array?

a. An array is allowed to contain duplicate values.
b. An array expands automatically when it is full.
c. An array contains objects of the same data type.
d. An array uses a zero index to reference the first element.

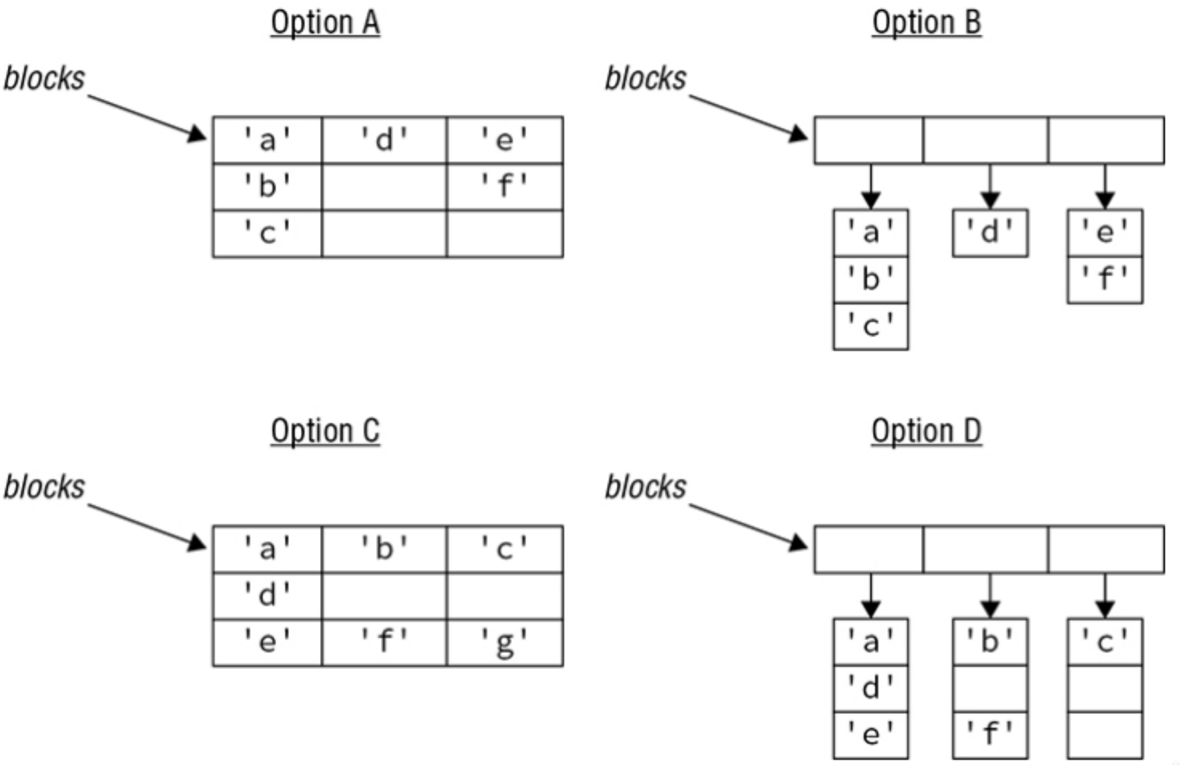
32. Which line of code causes an **ArrayIndexOutOfBoundsException**?

a. m1
b. m2
c. m3
d. m4

33. Which of the options in the graphic best represent the **blocks** variable?

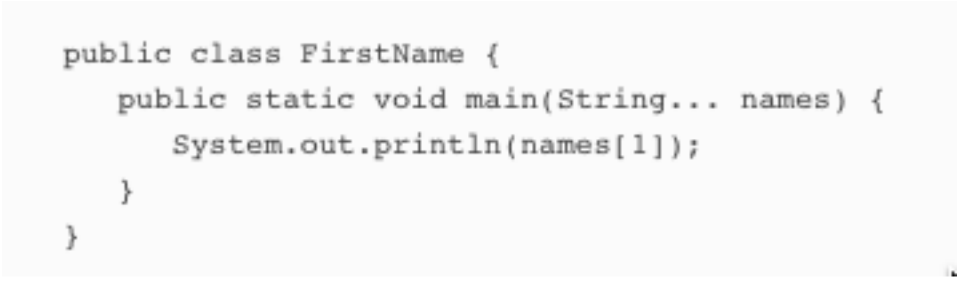


```
char[][] blocks = new char[][] { { 'a', 'b', 'c' }, { 'd' }, { 'e', 'f' } };
```



a. Option A
b. Option B
c. Option C
d. Option D

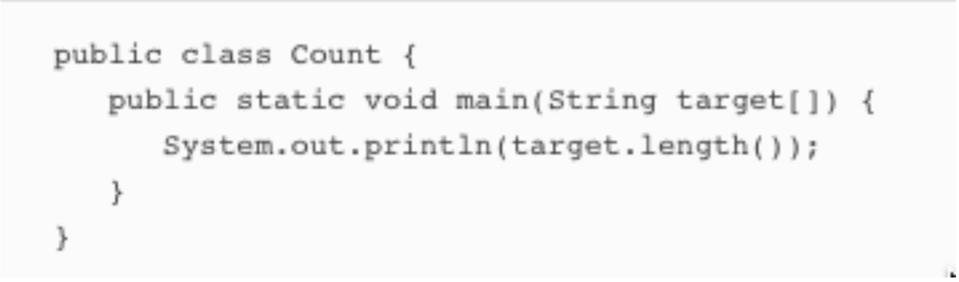
34. What is the output of the following when run as **java FirstName Wolfie** ?



```
public class FirstName {
   public static void main(String... names) {
      System.out.println(names[1]);
   }
}
```

a. FirstName
b. Wolfie
c. The code throws an ArrayIndexOutOfBoundsException.
d. The code throws a NullPointerException.

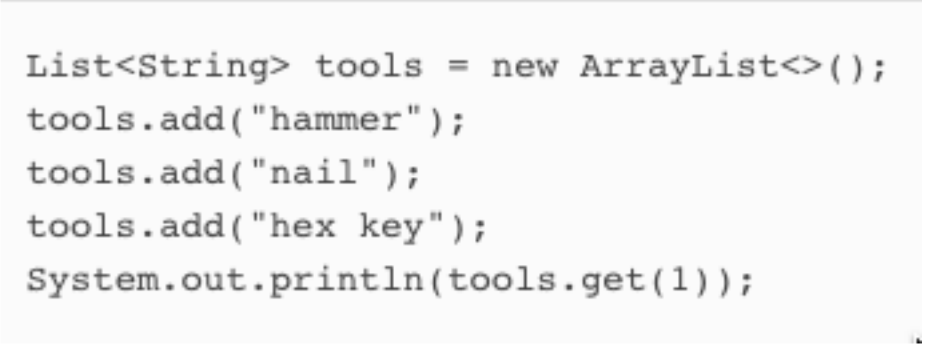
35. What is the output of the following when run as **java Count 1 2** ?



```
public class Count {
   public static void main(String target[]) {
      System.out.println(target.length());
   }
}
```

a. 0
b. 1
c. 2
d. The code does not compile.

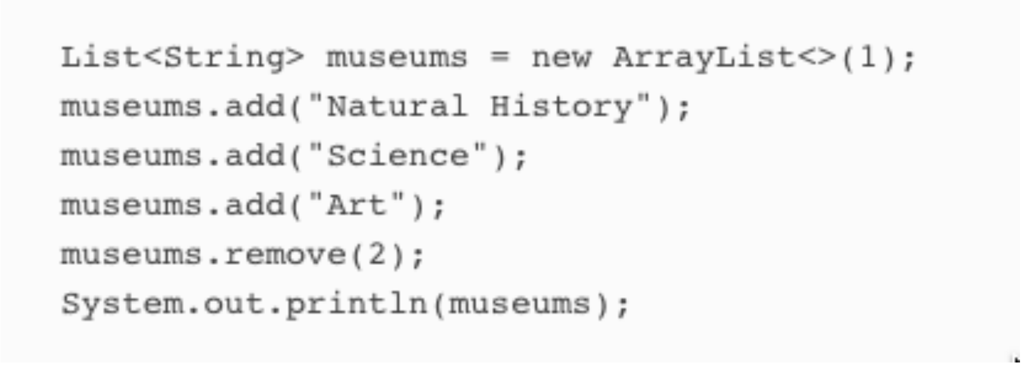
36. What is the output of the following?



```
List<String> tools = new ArrayList<>();
tools.add("hammer");
tools.add("nail");
tools.add("hex key");
System.out.println(tools.get(1));
```

a. hammer
b. hex key
c. nail
d. None of the above

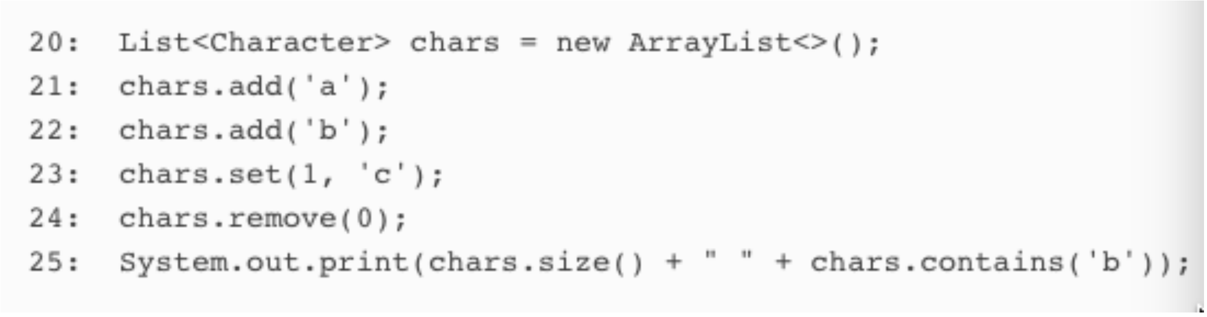
37. What is the output of the following?



```
List<String> museums = new ArrayList<>(1);
museums.add("Natural History");
museums.add("Science");
museums.add("Art");
museums.remove(2);
System.out.println(museums);
```

a. [Natural, History, Science]
b. [Natural, History, Art, Science]
c. The code does not compile.
d. The code compiles but throws an exception at runtime.

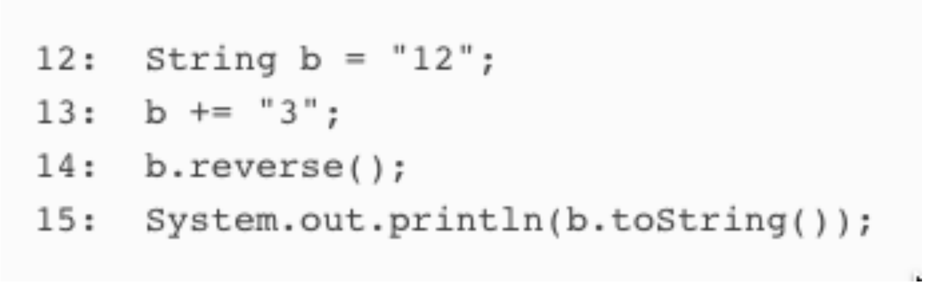
38. What is the output of the following?



```
List<Character> chars = new ArrayList<>();
chars.add('a');
chars.add('b');
chars.set(1, 'c');
chars.remove(0);
System.out.print(chars.size() + " " + chars.contains('b'));
```

a. 1 false
b. 1 true
c. 2 false
d. 2 true

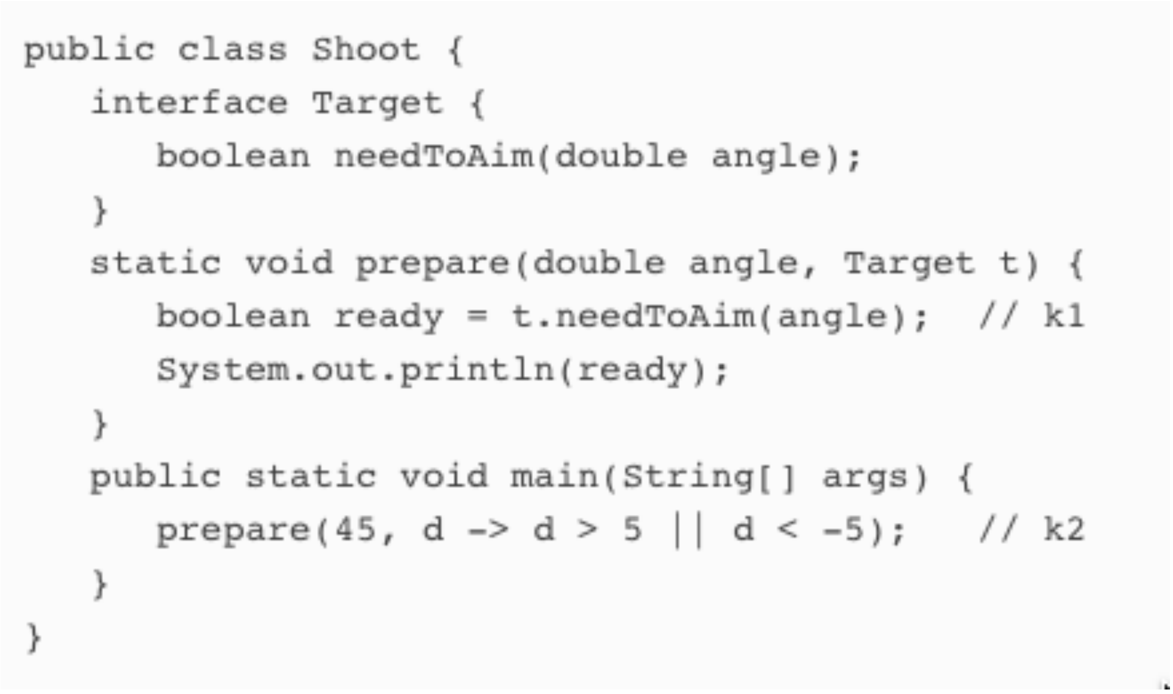
39. What is the output of the following?



```
String b = "12";
b += "3";
b.reverse();
System.out.println(b.toString());
```

a. 12
b. 123
c. 321
d. The code does not compile.

40. What does the following do?



```
public class Shoot {
   interface Target {
      boolean needToAim(double angle);
   }
   static void prepare(double angle, Target t) {
      boolean ready = t.needToAim(angle);  // k1
      System.out.println(ready);
   }
   public static void main(String[] args) {
      prepare(45, d -> d > 5 || d < -5);   // k2
   }
}
```

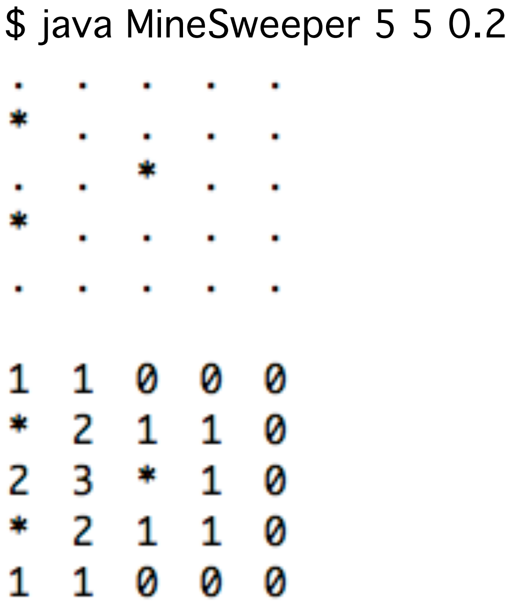
a. It prints **true**.
b. It prints **false**.
c. It doesn't compile due to line **k1**.
d. It doesn't compile due to line **k2.**

41. Write a Java program that use *recursion* to compute x^y (x raises to the power of y where x and y are integer). The signature of the function is as follows.

public static int power(int x, int y);

42. Write a Java program that takes 3 command line parameter M, N, p and produces an M-by-N Boolean array where each entry is occupied with probability p. In the minesweeper game, occupied cells represent bombs and empty cells represent safe cells. Print out the array using an asterisk for bombs and a period for safe cells. Then, replace each safe square with the number of neighboring bombs (above, below, left, right, or diagonal) and print out the solution.

For example, here is an example run of the Minesweeper program with **M=5, N=5, and p=0.2**



```
$ java MineSweeper 5 5 0.2
.  .  .  .  .
*  .  .  .  .
.  .  *  .  .
*  .  .  .  .
.  .  .  .  .

1  1  0  0  0
*  2  1  1  0
2  3  *  1  0
*  2  1  1  0
1  1  0  0  0
```