

## Question - 1

How many lines does this code output?

```
List<String> list = new LinkedList<>();  
list.add("Archie");  
list.add("X-Men");  
Stream<String> s = list.stream();  
s.forEach(System.out::println);  
s.forEach(System.out::println);
```

- A. Two
- B. Four
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.

## Question - 2

We want this code to print the titles of each book twice. Why doesn't it?

```
LinkedList<String> list = new LinkedList<>();  
list.add("Grapes of Wrath");  
list.add("1984");  
list.stream().forEach(System.out::println);  
Iterator it = list.iterator();  
while (it.hasNext())  
System.out.println(it.next());
```

- A. The generic type of `Iterator` is missing.
- B. The `hasNext()` method should be changed to `isNext()`.
- C. The iteration code needs to be moved before the `forEach()` since the stream is used up.
- D. None of the above. The code does print each book title twice.

### Question - 3

Rewrite this lambda using a method reference:

`() -> Math.random()`

A. `Math.random`

B. `Math::random`

C. `Math::random()`

D. None of the above

#### Question - 4

Which operation can occur more than once in a stream pipeline?



- A. Source
- B. Intermediate operation
- C. Terminal operation
- D. None of the above

### Question - 5

Which type allows inserting a `null` value?

A. ArrayDeque

B. ArrayList

C. TreeSet

D. All of these allow nulls.

## Question - 6

Fill in the blank so this code outputs three lines:

```
List<String> list = new ArrayList<>();  
list.add("Atlanta");  
list.add("Chicago");  
list.add("New York");  
list.stream().filter(_____).forEach(System.out::println);
```

- A. `String::isEmpty`
- B. `!String::isEmpty`
- C. `String::! isEmpty`
- D. None of the above

## Question - 7

What is the output of the following?

```
TreeMap<String, Integer> map = new TreeMap<>();  
map.put("3", 3);  
map.put("three", 3);  
map.put("THREE", 3);  
System.out.println(map.firstKey() + " " + map.lastKey());
```

- A. 3 three
- B. 3 THREE
- C. three 3
- D. THREE 3

## Question - 8

What is the result of the following?

```
List<String> list = new ArrayList<>();  
list.add("Austin");  
list.add("Boston");  
list.add("San Francisco");  
list.removeIf(a -> a.length() > 10);  
System.out.println(list.size());
```

- A. 1
- B. 2
- C. 3
- D. None of the above



## Question - 9

What does the following output?

```
ArrayDeque<Integer> dice = new ArrayDeque<>();  
dice.offer(3);  
dice.offer(2);  
dice.offer(4);  
System.out.print(dice.stream().filter(n -> n != 4));
```

- A. 2
- B. 3
- C. The code does not compile.
- D. None of the above

## Question - 10

What is the output of the following?

```
class Magazine implements Comparable<Magazine> {
    private String name;
    public Magazine(String name) {
        this.name = name;
    }
    @Override
    public int compareTo(Magazine m) {
        return name.compareTo(m.name);
    }
    @Override
    public String toString() {
        return name;
    }
}

public class Newstand {
    public static void main(String[] args) {
        Set<Magazine> set = new TreeSet<>();
        set.add(new Magazine("highlights"));
        set.add(new Magazine("Newsweek"));
        set.add(new Magazine("highlights"));
        System.out.println(set.iterator().next());
    }
}
```

- A. highlights
- B. Newsweek
- C. The code does not compile.
- D. The code compiles but throws an exception at runtime.