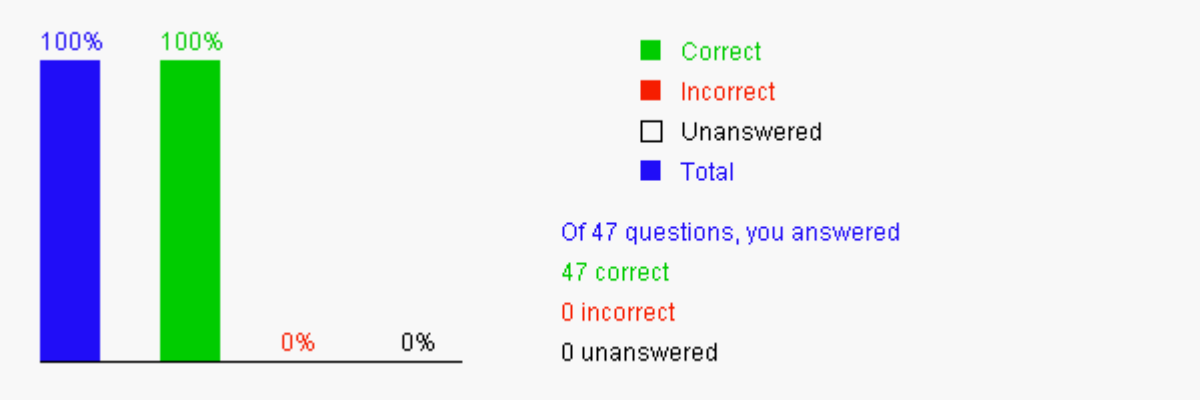
This quiz is for students to practice. A large number of additional quiz is available for instructors using Quiz Generator from the Instructor's Resource Website. Videos for Java, Python, and C++ can be found at https://yongdanielliang.github.io/revelvideos.html.

Chapter 30 Aggregate Operations on Collection Streams



Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate which book and edition you are using. Thanks!

Section 30.2 Stream Pipelines

30.1 The default _____ method is defined the Collection interface for creating a stream.

- A. getStream()
- B. Stream()
- C. stream()
- D. findStream()

Your answer is correct

30.2 The Stream interface extends _____.

- A. Collection
- B. Collections
- C. BaseStreams
- D. BaseStream

Your answer is correct

30.3 A stream pipeline may contain _____ intermediate methods.

- A. 0
- B. 1
- C. 0 or 1
- D. 0 or more

Your answer is correct

30.4 A stream pipeline may contain _____ terminal methods.

- A. 0
- B. 1
- C. 0 or 1
- D. 0 or more

Your answer is correct

30.5 The Stream<T> interface contains static methods _____ for creating instances of Stream.

- [x] A. empty()
- [x] B. of(T... values)
- [x] C. of(values: T)
- [x] D. concat(Stream<? extedns T> s1, Stream<? extends T> s2)

Your answer is correct

30.6 _____ is an intermediate method.

- [x] A. distinct
- [x] B. filter

- [x] C. limit
- [x] D. skip

Your answer is correct

30.7 _____ is an intermediate method.

- [x] A. sorted
- [x] B. map
- [x] C. mapToInt
- [x] D. mapToLong
- [x] E. mapToDouble

Your answer is correct

30.8 _____ is a terminal method.

- [x] A. count
- [x] B. max
- [x] C. min
- [x] D. findFirst
- [x] E. findAny

Your answer is correct

30.9 _____ is a terminal method.

- [x] A. allMatch
- [x] B. anyMatch
- [x] C. noneMatch
- [x] D. forEach

Your answer is correct

30.10 _____ is a terminal method.

- [x] A. reduce
- [x] B. collect
- [x] C. toArray
- [ ] D. map

Your answer is correct

30.11 Which of the following methods takes an argument of the Comparator type?

- [x] A. max
- [x] B. min
- [x] C. sorted
- [ ] D. forEach

Your answer is correct

30.12 The statement forEach(System.out::println) is the same as _____.

- ( ) A. forEach(System.out::print)
- ( ) B. forEach(System.out.println)
- ( ) C. forEach(e -> System.out.println())
- (•) D. forEach(e -> System.out.println(e))

Your answer is correct

30.13 The statement sorted((s1, s2) -> s1.compareToIgnoreCase(s2)) is the same as _____.

- ( ) A. sorted(String::compareToIgnoreCase(s))
- ( ) B. sorted(String::compareToIgnoreCase())
- (•) C. sorted(String::compareToIgnoreCase)
- ( ) D. sorted(compareToIgnoreCase(s))

Your answer is correct

30.14 What is the output of the following code?

```
Character[] chars = {'D', 'B', 'A', 'C'};
Stream.of(chars).filter(e -> e > 'B').sorted()
  .forEach(System.out::print);
```

- A. DC
- B. CD
- C. ABCD
- D. DCBA

Your answer is correct

30.15 What is the output of the following code?

```
Character[] chars = {'D', 'B', 'A', 'C'};
System.out.println(Stream.of(chars).allMatch(e -> e > 'B') + " " +
  Stream.of(chars).noneMatch(e -> e > 'B'));
```

- A. true true
- B. true false
- C. false true
- D. false false

Your answer is correct

30.16 What is the output of the following code?

```
Character[] chars = {'D', 'B', 'A', 'C'};
System.out.println(Stream.of(chars).skip(2)
  .max(Character::compareTo) + " " +
  Stream.of(chars).skip(2)
  .min(Character::compareTo));
```

- A. C C
- B. A A
- C. A C
- D. C A

Your answer is correct

30.17 What is the output of the following code?

```
Character[] chars = {'D', 'B', 'A', 'C'};
Stream.of(chars).map(e -> e.toLowerCase()).forEach(System.out::print);
```

- A. D B A C
- B. d b a c
- C. DBAC
- D. dbac

Your answer is correct

30.18 What is the output of the following code?

```
System.out.println(Stream.of(new Loan(2.5, 1, 10.0),
  new Loan(7.5, 5, 10.1), new Loan(7.5, 3, 10.2),
  new Loan(7.5, 3, 10.3))
  .max((e1, e2) -> e1.getNumberOfYears() - e2.getNumberOfYears())
  .get().getLoanAmount());
```

- A. 10.0
- B. 10.1
- C. 10.2
- D. 10.3

Your answer is correct

Section 30.3 IntStream, LongStream, and DoubleStream

30.19 _____ is a subtype of BaseStream.

- [x] A. Stream
- [x] B. IntStream
- [x] C. LongStream
- [x] D. DoubleStream

Your answer is correct

30.20 _____ represents a sequence of _____ values.

- [x] A. IntStream, int
- [x] B. LongStream, long
- [x] C. DoubleStream, double
- [ ] D. FloatStream, float

Your answer is correct

30.21 sum() is a method defined in _____.

- [ ] A. Stream
- [x] B. IntStream
- [x] C. LongStream
- [x] D. DoubleStream

Your answer is correct

30.22 What is the output of the following code?

```
Character[] chars = {'D', 'B', 'A', 'C'};
System.out.println(Stream.of(chars).mapToInt(e -> e - 'A').sum());
```

- ( ) A. 4
- ( ) B. 5
- (•) C. 6
- ( ) D. 7

Your answer is correct

30.23 What is the output of the following code?

```
double[] numbers = {1.2, 1, 2.2, 3.6};
System.out.println(DoubleStream.of(numbers)
  .mapToInt(e -> (int)e).sum());
```

- ( ) A. 4
- ( ) B. 5
- ( ) C. 6
- (•) D. 7

Your answer is correct

30.24 What is the output of the following code?

```
double[] numbers = {1.2, 1.23, 2.2, 3.6};
System.out.println(DoubleStream.of(numbers)
  .mapToObj(e -> e + "").mapToInt(e -> e.length()).sum());
```

- ( ) A. 12
- (•) B. 13
- ( ) C. 14
- ( ) D. 15

Your answer is correct

Section 30.4 Parallel Streams

30.25 To create a parallel stream from a Collection object c, use _____.

- [ ] A. c.stream()
- [x] B. c.parallelStream()
- [x] C. c.stream().parallel()
- [ ] D. c.stream().sequential()

Your answer is correct

30.26 The following statement displays _____.

```
IntStream.of(1, 2, 3, 4, 5).parallel()
  .forEach(e -> System.out.print(e + " "));
```

- ○ A. 1 2 3 4 5
- ○ B. 5 4 3 2 1
- ○ C. 1 2 3 5 4
- ◉ D. 1 2 3 4 5 in a random order

Your answer is correct 

30.27 _____ is a stateless method.

- ☐ A. sorted
- ☑ B. filter
- ☑ C. limit
- ☑ D. map
- ☐ E. distinct

Your answer is correct 

30.28 _____ is a stateful method.

- ☑ A. sorted
- ☐ B. filter
- ☐ C. limit
- ☐ D. map
- ☑ E. distinct

Your answer is correct 

Section 30.5 Stream Reduction Using the reduce method

30.29 The following statement displays _____.

```
System.out.println(IntStream.of(1, 2, 3, 4, 5).parallel()
  .reduce(0, (e1, e2) -> e1 + e2));
```

- ○ A. 14
- ◉ B. 15
- ○ C. 16
- ○ D. 17
- ○ E. 18

Your answer is correct 

30.30 The following statement displays _____.

```
System.out.println(IntStream.of(1, 2, 3, 4, 5).parallel()
  .reduce(1, (e1, e2) -> e1 * e2));
```

- ◉ A. 120
- ○ B. 140
- ○ C. 130
- ○ D. 150
- ○ E. 170

Your answer is correct 

30.31 The following statement displays _____.

```
System.out.println(IntStream.of(1, 2, 3, 4, 5).parallel()
  .reduce(Integer.MAX_VALUE, (e1, e2) -> Math.min(e1, e2)));
```

- ◉ A. 1
- ○ B. 2
- ○ C. 3
- ○ D. 4
- ○ E. 5

Your answer is correct 

30.32 The following statement displays _____.

```
System.out.println(IntStream.of(1, 2, 3, 4, 5)
  .mapToObj(e -> e + "").reduce((e1, e2) -> e1 + " " + e2).get());
```

- ○ A. nothing

- B. 1
- C. 2
- D. 3
- E. 1 2 3 4 5

Your answer is correct 

30.33 The following statement displays _____.

```
System.out.println(IntStream.of(1, 2, 3, 4, 5)
  .mapToObj(e -> e + "").reduce((e1, e2) -> e1 + " " + e2).get());
```

- A. nothing
- B. 1
- C. 2
- D. 3
- E. 1 2 3 4 5

Your answer is correct 

30.34 Show the output of the following code:

```
import java.util.stream.IntStream;
import java.util.stream.Stream;

public class Test {
   public static void main(String [] args){
    int[][] m = {{1, 2, 3}, {3, 4, 5}, {5, 2}, {1, 3}};
    Stream.of(m).map(e -> IntStream.of(e))
      .reduce((e1, e2) -> IntStream.concat(e1, e2))
      .get().distinct()
      .forEach(e -> System.out.print(e + " "));
   }
}
```

- A. 1 2 3 3 4 5 5 2 1 2
- B. 1 2 3 4 5
- C. 1 2 3 4 5 in a random order
- D. 5 4 3 2 1

Your answer is correct 

30.35 Show the output of the following code:import java.util.stream.IntStream;

```
import java.util.stream.Stream;

public class Test {
   public static void main(String [] args){
    int[][] m = {{1, 2, 3}, {3, 4, 5}, {5, 2}, {1, 3}};
    System.out.println(
      Stream.of(m).map(e -> IntStream.of(e))
        .reduce((e1, e2) -> IntStream.concat(e1, e2))
        .get().distinct().mapToObj(e -> e + "")
        .reduce((e1, e2) -> e1 + ", " + e2).get());
   }
}
```

- A. 1, 2, 3, 3, 4, 5, 5, 2, 1, 2
- B. 1, 2, 3, 3, 4, 5, 5, 2, 1, 2,
- C. 1, 2, 3, 4, 5,
- D. 1, 2, 3, 4, 5

Your answer is correct 

Section 30.6 Stream Reduction Using the collect Method

30.36 The following statement displays _____.

```
System.out.println(IntStream.of(1, 2, 3, 4, 5)
  .collect(() -> new ArrayList(), (c, e) -> c.add(e),
    (c1, c2) -> c1.addAll(c2)));
```

- A. [1, 2, 3, 4]
- B. [1]
- C. [1, 2]
- D. [1, 2, 3]
- E. [1, 2, 3, 4, 5]

Your answer is correct

30.37 The following statement displays _____.

```
System.out.println(IntStream.of(2, 5, 5, 4, 5)
  .collect(() -> new TreeSet(), (c, e) -> c.add(e),
    (c1, c2) -> c1.addAll(c2)));
```

- A. [2, 4, 5, 5]
- B. [2, 5, 4]
- C. [2, 5, 5, 4, 5]
- D. [5, 2, 4]
- E. [2, 4, 5]

Your answer is correct

30.38 The following statement displays _____.

```
System.out.println(IntStream.of(2, 5, 5, 4, 5)
  .mapToObj(e -> e).collect(Collectors.toList()));
```

- A. [2, 4, 5, 5]
- B. [2, 5, 4]
- C. [2, 5, 5, 4, 5]
- D. [5, 2, 4]
- E. [2, 4, 5]

Your answer is correct

30.39 The following statement displays _____.

```
IntStream.of(2, 5, 5, 4, 5)
  .mapToObj(e -> e).collect(Collectors.toSet())
  .stream().sorted().forEach(e -> System.out.print(e + " "));
```

- A. 2 4 5 5
- B. 2 5 4
- C. 2 5 5 4 5
- D. 5 2 4
- E. 2 4 5

Your answer is correct

Section 30.7 Grouping Elements Using the groupingby Collector

30.40 The following statement displays _____.

```
IntStream.of(1, 2, 3, 4, 1, 3, 5, 4, 4).mapToObj(e -> e).collect(
  Collectors.groupingBy(e -> e, Collectors.counting()))
  .forEach((k, v) -> {if (k == 3) System.out.println(k + " " + v);});
```

- A. 3 1
- B. 3 2
- C. 3 3
- D. 2 3
- E. 2 1

Your answer is correct

30.41 _____ is a terminal method.

- [x] A. reduce
- [x] B. collect
- [x] C. forEach
- [ ] D. map
- [ ] E. mapToInt

Your answer is correct

30.42 The forEach method is defined in the _____ interface.

- [x] A. Stream
- [x] B. IntStream

- [x] C. LongStream
- [x] D. DoubleStream
- [x] E. Collection

Your answer is correct 

30.43 The following statement displays _____.

```
IntStream.of(1, 2, 2, 1, 1).mapToObj(e -> e).collect(
  Collectors.groupingBy(e -> e, TreeMap::new, Collectors.counting()))
  .forEach((k, v) -> {System.out.print(k + " " + v + " ");});
```

- A. 1 3 2 3
- B. 1 3 2 1
- C. 1 3 2 2
- D. 2 3 1 3
- E. 2 2 1 3

Your answer is correct 

30.44 The following statement displays _____.

```
Map<Double, Double> map = Stream.of(new Loan(2.5, 1, 10.0),
  new Loan(7.5, 5, 10.1), new Loan(7.5, 3, 10.2),
  new Loan(7.5, 5, 10.3))
  .collect(Collectors.groupingBy(Loan::getAnnualInterestRate,
    TreeMap::new,
    Collectors.summingDouble(Loan::getLoanAmount)));
map.forEach((k, v) -> System.out.print(k + " " + v + " "));a. 1 3 2 3
```

- A. 7.5 10.0 2.5 30.6
- B. 7.5 30.6 2.5 10.0
- C. 2.5 7.5 2.5 30.6
- D. 2.5 10.0 7.5 30.6

Your answer is correct 

30.45 Show the output of the following code.

```
String s = "good";
Stream.of(s).forEach(e ->
  System.out.print(e + " "));
```

- A. g o o d
- B. good
- C. a random object reference
- D. [g, o, o, d]

Your answer is correct 

30.46 Show the output of the following code.

```
String s = "good";
Stream.of(s.toCharArray()).forEach(e ->
  System.out.print(e + " "));
```

- A. g o o d
- B. good
- C. a random object reference
- D. [g, o, o, d]

Your answer is correct 

30.47 Show the output of the following code.

```
import java.util.stream.Stream;

public class Test {
  public static void main(String[] args) {
    String s = "good";
    Stream.of(toCharacterArray(s.toCharArray())).forEach(e ->
      System.out.print(e + " "));
  }

  public static Character[] toCharacterArray(char[] list) {
    Character[] result = new Character[list.length];
    for (int i = 0; i < result.length; i++) {
```

```
      result[i] = list[i];
    }
    return result;
  }
}
```

- (●) A. g o o d
- ( ) B. good
- ( ) C. a random object reference
- ( ) D. [g, o, o, d]

Your answer is correct