

Given

```
1. public class Whiz{  
2.  
3.     static int i = 10;  
4.  
5.     public class Inner{  
6.         public static void print(){  
7.             System.out.println(i);  
8.         }  
9.     }  
10.  
11.    public static void main(String[] args){  
12.        Inner in = new Whiz().new Inner();  
13.        in.print();  
14.    }  
15. }
```

What is the output?

Please select :

- A. 10
- B. An Exception
- C. Compilation fails due to line 6
- D. Compilation fails due to line 7
- E. Compilation fails due to multiple errors

Given

```
public class Whiz {  
  
    static int y = 1;  
  
    public static void main(String[] args){  
  
        abstract class Cal{  
            public abstract int calc(int x);  
        }  
  
        Cal c = new Cal(){  
            public int calc(int x){return x-y;}  
        };  
  
        System.out.println(c.calc(2));  
    }  
}
```

What is the output?

Please select :

- A. 1
- B. 2
- C. An Exception is thrown.
- D. Compilation fails due to error at line 12.
- E. Compilation fails due to multiple errors.

Given :

```
1. interface I{
2.         default boolean equals(Object O){
3.             return true;
4.         }
5.     }
6.
7. class A implements I{
8.     public boolean equals(Object O){
9.         return false;
10.    }
11. }
12.
13. public class Whiz{
14.     public static void main(String [] args){
15.         A a = new A();
16.         I ia = new A();
17.         I i = new I();
18.
19.         System.out.println(a.equals(ia) + " " + i.equals(ia));
20.     }
21. }
```

What is the output?

Please select :

- A. true true
- B. false false
- C. true false
- D. An Exception.
- E. Compilation fails

Given

```
1. interface I{
2.     public default void print(){
3.         System.out.print("I");
4.     }
5.
6.     static void method(){
7.         System.out.print("Static");
8.     }
9. }
10.
11.
12.
13. public class Whiz{
14.     public static void main(String [] args){
15.         I i = new I();
16.         i.print();
17.         i.method();
18.     }
19. }
```

What is the output?

Please select :

- A. IStatic
- B. An exception.
- C. Compilation fails due to line 2.
- D. Compilation fails due to line 15.
- E. Compilation fails due to multiple errors.

Given following interface

```
interface Function<T>{  
    T apply(T t);  
}
```

Which of the following lambda expression can be used for change functional method?

Please select :

- A. String::length;
- B. String::toUpperCase;
- C. s -> return s;
- D. ()-> "ABC";
- E. Any of above.

Given

```
1. public interface A<R> extends B{  
2. }  
3.  
4. public interface B<T> {  
5.     public void print(T t);  
6. }  
7.  
8. public interface C<T> extends A,B{  
9.     public void printer(T t);  
10. }
```

Which of the interfaces can be considered as functional interface/s?

Please select :

- A. Only interface A.
- B. Only interface B.
- C. Only interfaces A and B.
- D. Only interfaces B and C.
- E. A, B and C, all three interface are functional interface

Given

```
1. public class WhizLab {  
2.     public static void main(String[] args) {  
3.         Value[] values = { Value.LOW, Value.HIGH, Value.MEDIUM };  
4.         Arrays.sort(values);  
5.         System.out.println(values[1]);  
6.     }  
7. }  
8.  
9.  
10.  
11. enum Value {  
12.     HIGH(1), MEDIUM(6), LOW(3);  
13.  
14.     private final int level;  
15.  
16.     private Value(int levelCode) {  
17.         level = levelCode;  
18.     }  
19. }
```

What is the output?

Please select :

- A. LOW
- B. MEDIUM
- C. HIGH
- D. An Exception.
- E. Compilation fails.

Which of the following services provided by the JDBC?

Please select :

- A. Creating a database management system.
- B. Retrieve the results received from the database in answer to your query but not to process them.
- C. Create, Send queries and update statements to the database
- D. It provides a database management system.
- E. None of above.

Given

```
1. import java.sql.*;
2.
3. public class Whiz{
4.     public static void main(String []args){
5.         DBConnect dc = new DBConnect();
6.     }
7. }
8. class DBConnect{
9.     private Connection con;
10.    public DBConnect(){
11.        try{
12.            Class.forName("com.mysql.jdbc.Driver");
13.            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/whiz""root""");
14.        }catch(Exception e){
15.            System.out.print("exception");
16.        }
17.    }
18. }
```

If the above code executes fine without printing “exception”, which of the followings can be considered as true? (Choose 2)

Please select :

- A. There is a table called whiz.
- B. There is a table called localhost.
- C. There is a database called whiz.
- D. There is no password for the user “root” in the working mysql server instance.

Which of the following connection URLs are valid?

- I. "jdbc:mysql://localhost:3306/whiz?"+"user=root&password=whizlabs"
- II. "jdbc:mysql://:3306/whiz?user=root&password=whizlabs"
- III. "jdbc:mysql://localhost:3306/whiz","root","whizlabs"

Please select :

- A. Only I.
- B. Only II.
- C. Only III.
- D. Only II and III.
- E. Only I and II.
- F. All.

Given

```
1. try{
2.     con.setAutoCommit(false);
3.
4.     Savepoint sp1 = con.setSavepoint("save1");
5.
6.         String query1 = "DELETE FROM allstar WHERE ID = 103";
7.         stm.executeUpdate(query1);
8.
9.         Savepoint sp2 = con.setSavepoint("save2");
10.
11.        String query2 = "DELETE FROM allstar WHERE ID = 124";
12.        stm.executeUpdate(query2);
13.
14.        con.rollback(sp2);
15.
16.        con.commit();
17.
18. }finally{
19.     con.close();
20. }
```

Note: Assume "con" is active connection and "stm" as a reference to valid Statement object. And consider the following snapshot of the table "allstar". Assume the method which this code fragment belongs handle the checked exception.

| ID | FName | LName | Birthday | Country |
|-----|------------|--------------|------------|--------------|
| 58 | Stuart | Broad | 1986-06-24 | England |
| 89 | Herschelle | Gibbs | 1974-02-23 | South Africa |
| 103 | Lou | Vincent | 1978-11-11 | New Zealand |
| 115 | Shahid | Afridi | 1980-03-01 | Pakistan |
| 124 | Sanath | Jayasuriya | 1969-06-30 | Sri Lanka |
| 136 | Lou | Vincent | 1978-11-11 | New Zealand |
| 178 | Piyush | Chawla | 1988-12-24 | India |
| 248 | Ryan | tenDoeschate | 1980-06-30 | Netherlands |

Which of the following is true after executing the above code fragment?

Please select :

- A. There will be no changes to the "allstar" table.
- B. Only the record with "ID" 103 will be deleted.
- C. Only the record with "ID" 124 will be deleted.
- D. Compilation will be failed if we added this code fragment to a program.
- E. An exception will be thrown at runtime.

Given

```
1. import java.io.*;
2. class NotInRangeException extends IOException{
3.     NotInRangeException(){
4.         super("Not in valid Range");
5.     }
6. }
7.
8. class Employee{
9.     int age;
10.    public boolean setAge(int ae) throws NotInRangeException{
11.        if(ae < 18 || ae > 60){
12.            throw new NotInRangeException();
13.        }else{
14.            age = ae;
15.            return true;
16.        }
17.    }
18. }
19.
20. public class Whiz{
21.     public static void main(String [] args) throws NotInRangeException{
22.         Console con = System.console();
23.         Employee d = new Employee();
24.         int i = Integer.parseInt(con.readLine("Enter the age : "));
25.         d.setAge(i);
26.         System.out.print(d.age);
27.     }
28. }
```

Which is true?

Please select :

- A. There is no constructor in `java.lang.IOException` class which takes a string.
- B. We can't create custom exception by extending `java.io.IOException` class.
- C. The “`NotInRangeException`” is an unchecked Exception since it doesn't extend `java.lang.Exception` class.
- D. Compilation succeeds and if we enter a number greater than 60, a “`NotInRangeException`” will be thrown and it'll look like “Exception in thread "main" `NotInRangeException: Not in valid Range`”.
- E. Compilation fails

Given

```
1. public class Whiz{  
2.     public static void main(String[] args) {  
3.         assert args == null : "No ";  
4.         System.out.print("Good");  
5.     }  
6. }
```

What is the output? (Assertion is enabled)

Please select :

- A. Good
- B. NO
- C. An AssertionError with message No
- D. An Exception
- E. Compilation fails

Which of the following is the correct method signature of the abstract method in AutoCloseable interface?

Please select :

- A. public void finally()
- B. public void exit()
- C. void exit() throws Exception
- D. public void close() throws Exception
- E. None of above.

Given

```
1. class Whiz{
2.     public static void main(String args[]){
3.         try{
4.             new Whiz().meth();
5.         }catch(Exception e){
6.             System.out.print("Exception");
7.         }catch(ArithmeticException e){
8.             System.out.print("Arithmatic");
9.         }
10.    }
11.
12.    public void meth()throws ArithmeticException{
13.        for(int x=0;x<5;x++){
14.            int y = (int)5/x;
15.            System.out.print(x);
16.        }
17.    }
18. }
```

What is the output?

Please select :

- A. Arithmetic
- B. An uncaught exception is thrown at runtime.
- C. Compilation fails.
- D. No output
- E. Exception

Given

```
1. class Ex6{
2.     public static void main(String args[]){
3.         try{
4.             new Ex6().meth();
5.         }catch(ArithmetricException e){
6.             System.out.print("Arithmetric");
7.         }finally{
8.             System.out.print("final 1");
9.         }catch(Exception e){
10.             System.out.print("Exception");
11.         }finally{
12.             System.out.print("final 2");
13.         }
14.     }
15.
16.     public void meth()throws ArithmetricException{
17.         for(int x=0;x<5;x++){
18.             int y = (int)5/x;
19.             System.out.print(x);
20.         }
21.     }
22. }
```

What is the output?

Please select :

- A. Arithmetric final 1
- B. Exception final 2
- C. Arithmetric final 2
- D. Compilation fails.
- E. Arithmetric

Which of the following can be considered as a source for a Stream?

- I. A collection
- II. An array
- III. I/O resources

Please select :

- A. Only I
- B. Only III
- C. Only I and II
- D. Only II and III
- E. All.

Given

```
1. import java.util.List;
2. import java.util.ArrayList;
3.
4. public class Whiz {
5.     public static void main(String[] args) {
6.         List<String> lst = new ArrayList<>();
7.         lst.add("a");
8.         lst.add("ab");
9.         lst.stream().forEach(s ->
10.             System.out.print(s.length()));
11. }
```

What is the output?

Please select :

- A. 1a2ab
- B. a1ab2
- C. No output.
- D. An Exception.
- E. Compilation fails

Which of the following lambda can be used for the forEach method to iterate through all elements of a map?

Please select :

- A. (k,v)->System.out.println(v)
- B. (k)->System.out.println(k)
- C. System.out::println
- D. (k,v)-> k;
- E. None of above.

Given

```
1. import java.util.*;
2. class Dog implements Comparable<Dog>{
3.     private String name;
4.     private int age;
5.     private String owner;
6.     Dog(String n, String o, int a){
7.         name = n;          owner = o;          age = a;
8.     }
9.     public String toString(){
10.        return name;
11.    }
12.    public int compareTo(Dog d){
13.        return owner.compareTo(d.owner);
14.    }
15. }
16. public class Dtest{
17.     public static void main(String [] args){
18.         ArrayList<Dog> doglist = new ArrayList<Dog>();
19.         doglist.add(new Dog("Lazy", "John", 3));
20.         doglist.add(new Dog("White", "Henry", 2));
21.         doglist.add(new Dog("Blaky", "Bert", 5));
22.         doglist.add(new Dog("Tazan", "Jack", 1));
23.         Collections.sort(doglist);
24.         System.out.print(doglist);
25.     }
26. }
```

What is the result?

Please select :

- A. [Lazy, White, Blaky, Tazan]
- B. [Blaky, White, Tazan, Lazy]
- C. [Bert, John, Jack, Henry]
- D. [Bert, Henry, Jack, John]
- E. [Blaky, Lazy, Tazan, White]

Given

```
1. import java.util.stream.Stream;
2.
3. public class Whiz{
4.
5.     public static void main(String[] args) {
6.         Stream<String> stream = Stream.of("A","AB","ABC","ABCD","AB");
7.         String out = stream.filter(s->s.length()>2).
8.                         filter(s->s.indexOf("C")>-1).findAny().orElse("None");
9.         System.out.println(out);
10.
11.    }
```

What is the output?

Please select :

- A. None
- B. ABCD
- C. ABC
- D. An Exception.
- E. Compilation fails.

Given

```
1. import java.util.stream.IntStream;
2. import java.util.stream.Stream;
3.
4. public class Whiz{
5.         public static void main(String[] args) {
6.             IntStream ds = IntStream.of(1,2,2,4);
7.             Stream<String> ste = ds.boxed().map(Integer::toString);
8.             System.out.print(ste.distinct().findFirst());
9.         }
10. }
```

What is the output?

Please select :

- A. 1
- B. 2
- C. 4
- D. An Exception
- E. Compilation fails

Given

Comparator<Integer> comp = -----

Which of the following method reference can be used to fill the blank?

Please select :

- A. Integer:comparing;
- B. Integer::comparing();
- C. Integer::compare;
- D. Integer->sum()
- E. None of above.

Which of the following operation of java.util.stream package is stateful intermediate operation?

Please select :

- A. sorted.
- B. filter.
- C. max.
- D. peek.
- E. None of above

Given

```
1. import java.util.stream.Stream;
2.
3. public class Whiz{
4.
5.     public static void main(String[] args) {
6.         Stream<Integer> stream = Stream.of(4,10,8,3);
7.         System.out.println(stream.peek(System.out::print).anyMatch(x -> x>=10));
8.     }
9. }
```

What is the output?

Please select :

- A. true
- B. 41083true
- C. 410true
- D. An Exception.
- E. Compilation fails.

Given

```
1. import java.util.ArrayList;
2. import java.util.List;
3.
4. public class Whiz{
5.
6.     public static void main(String[] args){
7.         List<String> list = new ArrayList<>();
8.         list.add("and"); list.add("an");
9.         list.add("animal"); list.add("Ant");
10.        list.removeIf(e -> e.startsWith("an"));
11.        System.out.println(list);
12.    }
13.}
```

Which is the output?

Please select :

- A. [and, an, animal, Ant]
- B. [Ant]
- C. [an, Ant]
- D. Compilation fails due to error at line 7.
- E. Compilation fails due to error at line 10

Which of the following method in `java.util.Map` introduced in java SE 8?

Please select :

- A. `subList(int , int)`
- B. `removeAll(Collection)`
- C. `replaceAll(Object, V)`
- D. `getOrDefault(Object, V)`
- E. Any of above.

Given

```
1. import java.util.HashMap;
2. import java.util.Map;
3.
4. public class Whiz{
5.
6.     public static void main(String[] args) {
7.         Map<Integer,String> map = new HashMap<>();
8.         map.put(1, "A");
9.         map.put(2, "B");
10.        map.put(3, "C");
11.        map.put(4, "D");
12.        map.remove("A");
13.        map.remove(3,"C");
14.        map.remove(4,"B");
15.        map.values().forEach(System.out::print);
16.    }
17. }
```

What is the output?

Please select :

- A. ABD
- B. AD
- C. ABCD
- D. BCB
- E. Compilation fails.

Which of the following lambda can be used to represent Comparator for integer array?

Please select :

- A. Comparator.comparing(e -> e.size());
- B. Comparator.comparing(e -> e.length());
- C. e1, e2 ->Integer::compare;
- D. Comparator<String> c4 = (e1, e2) -> e1.length>e2.length;
- E. None of above.

Given

```
1. import java.util.ArrayList;
2. import java.util.List;
3.
4. public class Whiz{
5.
6.     public static void main(String args[]){
7.         List<? extends Number> list = new ArrayList<>();
8.         list.add(new Integer(10));
9.         list.add(new Double(1.2));
10.
11.        System.out.println(list.get(0)+ ""+list.get(1));
12.    }
13. }
```

What is the output?

Please select :

- A. 101
- B. 101.2
- C. 11.2
- D. Compilation fails due to error at line 7.
- E. Compilation fails due to multiple errors.

Given

```
1. class Ttest implements Runnable{
2.     public static void main(String [] args){
3.         Ttest a1= new Ttest();
4.         Thread t=new Thread(a1);
5.         t.setName("a1");
6.         t.start();
7.         System.out.print(Thread.currentThread().getName() + " ");
8.         try{
9.             t.join();
10. }catch(Exception e){}
11.         System.out.print(Thread.currentThread().getName());
12.     }
13.
14.     public void run(){
15.         System.out.print(Thread.currentThread().getName() + " ");
16.     }
17. }
```

What is the result? (Choose 2)

Please select :

- A. The output could be main main a1.
- B. The output could be main a1 main.
- C. The output could be a1 main main.
- D. The output could be main a1.
- E. An uncaught exception is thrown at runtime.

Which is true? (Choose 2)

Please select :

- A. Using the “start()” method, we can always enter a thread to the “Running” state, straight way.
- B. When Thread is dead it can start again by invoking start method again.
- C. By invoking the “stop()” method on a running thread, we can move it to “Runnable” state
- D. It is not guaranteed that every tread will take turns in fair way and it depends on virtual machine implementation.
- E. Once start method is invoked on a thread, it enters the “Runnable” state

Consider the following code.

```
1. class Sum extends RecursiveAction<Long> {
2.
3.     static final int MAX = 50000;
4.     int low;
5.     int high;
6.     int[] array;
7.
8.     Sum(int[] arr, int lo, int hi) {
9.         array = arr;
10.        low = lo;
11.        high = hi;
12.    }
13.
14.    public Long compute() {
15.
16.        if(high - low <= MAX) {
17.            long sum = 0;
18.            for(int i=low; i < high; ++i)
19.                sum += array[i];
20.            return sum;
21.        } else{
22.
23.            int mid = low + (high - low) / 2;
24.            Sum left = new Sum(array, low, mid);
25.            Sum right = new Sum(array, mid, high);
26.            long rightAns = right.compute();
27.            left.fork();
28.            long leftAns = left.join();
29.            return leftAns + rightAns;
30.        }
31.    }
32. }
```

Which of the following is true? (Choose 2)

Please select :

- A. Code will successfully compile.
- B. Compilation fails.
- C. If the number of elements in the array "arr" is larger than 50000, then code will divide the program in to small fork-join tasks.
- D. For this purpose we have to extend RecursiveTask instead of extending the RecursiveAction.
- E. For this purpose we could also extends the RecursiveTask.

Given

```
1. import java.util.concurrent.ForkJoinPool;
2. import java.util.concurrent.RecursiveTask;
3.
4. public class FT{
5.
6.     public static void main(String []args){
7.         int []a = new int[15000];
8.         for(int x = 0;x < 15000;x++){
9.             a[x] = 1;
10.        }
11.
12.        Sum s = new Sum(a,0,a.length);
13.        System.out.print(Sum.sumArray(a));
14.    }
15. }
16.
17. class Sum extends RecursiveTask<Long> {
18.
19.     static final int MAX = 5000;
20.
21.     int low;
22.     int high;
23.     int[] array;
24.
25.     Sum(int[] arr, int lo, int hi) {
26.         array = arr;
27.         low = lo;
28.         high = hi;
29.     }
30.
31.     public Long compute() {
32.
33.         if(high - low <= MAX) {
34.             long sum = 0;
35.             for(int i=low; i < high; ++i)
36.                 sum += array[i];
37.             return sum;
38.
39.         } else{
40.             int mid = low + (high - low) / 2;
41.             Sum left = new Sum(array, low, mid);
42.             Sum right = new Sum(array, mid, high);
43.             left.fork();
44.             long rightAns = right.compute();
45.             long leftAns = left.join();
46.             return leftAns + rightAns;
47.         }
48.     }
49.     static long sumArray(int[] array) {
50.         ForkJoinPool fPool = new ForkJoinPool();
51.         long sum = fPool.invoke(new Sum(array,0,array.length));
52.         return sum;
53.     }
54. }
```

Which of the following is true?

Please select :

- A. The output will be 0 and this is not the correct way of using the Fork and Join.
- B. The output will be 15000 and this is the correct way of using the Fork and Join.
- C. The output will be 15000 but this is not the correct way of using the Fork and Join.
- D. Compilation fails.
- E. An exception will be thrown at the runtime.

Given :

```
1. import java.util.*;
2. import java.util.stream.Collectors;
3.
4. public class WhizLab{
5.     public static void main(String[] args){
6.         List<Integer> list = Arrays.asList(new Integer[] { 3, 4, 5, 6, 7, 8, 9 });
7.         HashMap<Integer, List<Integer>> map = list
8.             .parallelStream().collect(Collectors.groupingByConcurrent(i -> i % 2 == 0 ? 1 : 2));
9.         System.out.println(map.get(0));
10.    }
11. }
```

What is the output?

Please select :

- A. Always [8, 4, 6]
- B. null
- C. It will print a list with 8, 4 and 6 as elements but in random order.
- D. An Exception.
- E. Compilation fails.

Given

```
1. import java.util.*;
2.
3. public class Whiz{
4.     public static void main(String[] args){
5.         String[] list = {"1","2","3"};
6.         Arrays.parallelSetAll(list,x->Integer.toString(x)+list[x]);
7.         System.out.println(list[0]);
8.     }
9. }
```

What is the output?

Please select :

- A. 10
- B. 11
- C. 01
- D. An Exception.
- E. Compilation fails.

Given

```
File f = new File("ocajp.txt");
```

Which of the following can be used to load above file content to a list?

Please select :

- A. Files.readAllLines(f.getAbsolutePath());
- B. Files.lines(f.getAbsolutePath());
- C. Files.readAllLines(f);
- D. Files.readAllLines(Paths.get(f.getAbsolutePath()));
- E. Files.lines(f);

Given

```
1. import java.io.IOException;
2. import java.nio.charset.Charset;
3. import java.nio.file.Files;
4. import java.nio.file.Paths;
5. import java.util.ArrayList;
6. import java.util.List;
7. import java.util.stream.Collectors;
8. import java.util.stream.Stream;
9.
10. public class Whiz{
11.     public static void main(String[] args) throws IOException{
12.         Stream<String> stream = Files.readAllLines(Paths.get("input.txt"));
13.         stream.skip(1).forEach(System.out::print);
14.     }
15. }
```

Content of the input.txt :

A

B

C

What is the output?

Please select :

- A. ABC
- B. AB
- C. BC
- D. Compilation fails due to error at line 12.
- E. Compilation fails due to error at line 13.

Given

```
1. import java.nio.file.*;
2. import java.io.*;
3. class NIO{
4.     public static void main(String [] args) throws IOException{
5.         Path path1 = Paths.get("F:\\Whizlabs\\java\\NIO");
6.         Path path2 = Paths.get("c:\\output");
7.         Path path = Paths.get(path2.getRoot().toString(), path1.subpath(0,3).toString());
8.         System.out.print(path.toString());
9.
10.    }
11. }
```

Note that "F:\\Whizlabs\\java\\NIO" actually exists while there is no "C:\\output" folder.

Which is the output?

Please select :

- A. c:\\Whizlabs\\java\\NIO
- B. c:\\Whizlabs\\java
- C. f:\\Whizlabs\\java\\NIO
- D. An IllegalArgumentException is thrown at the runtime
- E. Compilation fails as C:\\output" doesn't exist.

Given

```
1. import java.nio.file.*;
2. import java.io.*;
3. import static java.nio.file.StandardCopyOption.*;
4.
5. class NIO{
6.     public static void main(String [] args)throws IOException{
7.         Path path1 = Paths.get("F:\\Whizlabs\\java\\nio\\myfile.txt");
8.         Path path2 = Paths.get("..\\myfile.txt");
9.         Files.copy(path1, path2,REPLACE_EXISTING);
10.    }
11. }
```

Note that “F:\\Whizlabs\\java\\nio\\myfile.txt” actually exists but no other “myfile.txt” in any directories used in this code and current working directory is “F:\\Whizlabs\\java\\nio”. You have enough permission for above operations.

Which is true?

Please select :

- A. A file called “myfile.txt” will be created in the F:\\Whizlabs\\java\\nio after running the code first time.
- B. No file will be copied after running the code first time.
- C. Even If we try to run the code more than one time, a “FileAlreadyExistsExceptionexception” will not be thrown at the runtime.
- D. If we try to run the code more than one time, a “FileAlreadyExistsExceptionexception” will be thrown at the runtime.
- E. Compilation fails