

## 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 =
6.             Paths.get("F:\\Whizlabs\\java\\nio\\myfile.txt");
7.         Path path2 = Paths.get(".\\myfile.txt");
8.         System.out.print(Files.isSameFile(path1, path2));
9.     }
}
```

Note that "F:\\Whizlabs\\java\\nio\\myfile.txt" actually exists and current working directory is "F:\\Whizlabs\\java\\nio". You have enough permission for above operation.

Which is the output?

Please select :

- A. false
- B. true
- C. An exception will be thrown at the runtime.
- D. Compilation fails.

**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\\myfile.txt");
6.         Path path2 = Paths.get("myfiles\\myfile.txt");
7.         Path path = path2.resolve(path1);
8.         System.out.print(path.toString());
9.     }
10. }
```

**Note the current working directory is F:\\Whizlabs\\java\\NIO.**

**Which is the output?**

Please select :

- A. F:\\Whizlabs\\java\\NIO\\myfile.txt
- B. F:\\Whizlabs\\java\\NIO\\myfiles\\myfile.txt
- C. F:\\Whizlabs\\java\\NIO\\myfiles
- D. Compilation fails.
- E. An exception will be thrown at runtime.

## 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\\myfiles");
6.         Path path2 =
7.             Paths.get("F:\\Whizlabs\\java\\NIO\\myfiles\\data");
8.         //insert here
9.         System.out.print(path.toString());
10.    }
```

- I. Path path = path1.relativize(path2);
- II. Path path = path2.relativize(path1);

Which is true?

Please select :

- A. Inserting statement I at line 7, will produce the output as “..”
- B. Both, will produce the output as “..”
- C. Inserting statement II at line 7, will produce the output as “data”
- D. Both will cause compile time error.
- E. None of above

**Consider following path**

```
Path path = Paths.get("users\\whizlabs\\output");
```

**Which is correct?**

Please select :

- A. If we use "System.out.print(path.toAbsolutePath().toString());" the output will be "users\whizlabs\output"
- B. If we use "System.out.print(path.getNameCount());" the output will be 5.
- C. If we use "System.out.print(path.getName(2)) the output will be "whizlabs".
- D. If we use "System.out.print(path.getFileName());" the output will be "null".
- E. If we use "System.out.print(path.getRoot());"the output will be "null".

**Given :**

```
1. import java.io.IOException;
2. import java.nio.file.Files;
3. import java.nio.file.Path;
4. import java.nio.file.Paths;
5. import java.util.stream.Stream;
6.
7. public class Whizlab{
8.
9.     public static void main(String[] args) throws IOException {
10.
11.         Stream<Path> list = Files.list(Paths.get("F:\\ocpjp"));
12.         list.forEach(p ->System.out.print(p.getFileName() + " | "));
13.     }
14. }
```

**Content of the ocpjp directory**

**ocpjp**

- file.txt
- one[empty directory]
- two
  - o file2.txt
  - o three[empty directory]

**What is the output?**

Please select :

- A. file.txt | one | two | file2.txt | three
- B. one | two | three
- C. file.txt | one | two |
- D. An Exception.
- E. Compilation fails.

**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 Whizlab{
11.     public static void main(String[] args) throws IOException{
12.         Stream<String> stream = Files.readAllLines(Paths.get("whiz.txt"));
13.         stream.skip(1).forEach(System.out::print);
14.     }
15. }
```

**Content of the whiz.txt :**

AB  
A  
AB

**What is the output?**

Please select :

- A. ABAAB
- B. AAB
- C. ABA
- D. An exception.
- E. Compilation fails due to error at line 12.

**Given**

**File f = new File("whiz.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.*;
2. class Co{
3.
4.     public static void main(String args[])throws IOException{
5.         int i;
6.         InputStream in = System.in;
7.         BufferedReader br = new BufferedReader(new InputStreamReader(in));
8.         try{
9.             i = br.read();
10.            System.out.print("ASCII value of " + (char)i + " is "+ i);
11.        }catch(IOException e){
12.            System.out.print("IOException");
13.        }
14.    }
15. }
```

Which is true?

Please select :

- A. Compilation fails due to error on line 6.
- B. Compilation succeeds and if we enter 'B' at the prompt we get the output as, ASCII value of B is 66.
- C. In this code try/catch box is a must.
- D. Compilation fails due to error on line7.
- E. Compilation fails due to error on line 9.

**Given**

```
1. import java.io.*;
2.
3. class Co{
4.
5.     public static void main(String args[]){
6.         Console con = System.console();
7.         char []c = con.readPassword("Enter the password : ");
8.         System.out.write("Your password is : ");
9.         for(int i :c){
10.             System.out.write(i);
11.             System.out.flush();
12.         }
13.     }
14. }
```

**Note:** assume that you have access to console object.

**Which is true?**

Please select :

- A. Compilation succeeds and will prompt to enter a password and then it'll print the password.
- B. Compilation succeeds and will prompt to enter a password but then it won't print the password.
- C. Compilation fails as there is no "write()" method defined.
- D. Compilation fails due to error on line 8.
- E. Compilation fails due to error on line 10.

**Given :**

```
1. import java.util.Arrays;
2. import java.util.List;
3.
4. public class Whizlab{
5.
6.     public static void main(String[] args){
7.         List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 4, 6, 6);
8.         System.out.println( numbers.stream().filter(e -> e%2==0).mapToInt(i -> i).sum() );
9.     }
10. }
```

**What is the output?**

Please select :

- A. 26
- B. 22
- C. 0
- D. Compilation fails due to error at line 7.
- E. Compilation fails due to error at line 8.

**Given**

```
1. import java.util.function.Predicate;
2. import java.util.stream.Stream;
3.
4. public class Whizlab{
5.
6.     public static void main(String[] args){
7.
8.         Stream<Client> clients = Stream.of(new Client("Will","vps server",200),
9.                                         new Client("Rachel","java program",1200),new Client("Anthony","Configuration",1000));
10.
11.        Predicate<Client> func = e -> {   System.out.print( e.getName() + " " );
12.                                         return e.getBudget() > 1000.0;
13.                                         };
14.        clients.noneMatch(func);
15.    }
16. }
```

```
17.
18. class Client{
19.
20.     double budget;
21.     String name;
22.     String project;
23.
24.     Client(String n,String pr ,double p){
25.         budget = p;
26.         name = n;
27.         project = pr;
28.     }
29.
30.     public double getBudget(){
31.         return budget;
32.     }
33.
34.     public String getName(){
35.         return name;
36.     }
37.
38.     public String toString(){
39.         return name + "," + budget;
40.     }
41. }
```

**What is the output?**

Please select :

- A. Will Rachel
- B. Rachel
- C. Will Rachel Anthony
- D. Compilation fails due to error at line 11.
- E. Compilation fails due to error at line 14.

**Given**

```
1. import java.util.stream.DoubleStream;  
2.  
3. public class Whizlab{  
4.  
5.     public static void main(String[] args) {  
6.  
7.         DoubleStream doubles = DoubleStream.of(0.45,0.42,0.49);  
8.         System.out.print(doubles.filter(e -> e > 0.45).count());  
9.     }  
10. }
```

**What is the output?**

Please select :

- A. 1
- B. 2
- C. Compilation fails since we haven't imported predicate.
- D. Compilation fails due to error at line 7.
- E. Compilation fails due to error at line 8.

**Given**

```
1. import java.util.stream.IntStream;  
2.  
3. public class Whizlab{  
4.  
5.     public static void main(String[] args){  
6.  
7.         IntStream ints = IntStream.of(10,20,30);  
8.  
9.         ints.peek(d -> System.out.print(d*2 + " ")).count();  
10.    }  
11. }
```

**What is the output?**

Please select :

- A. **10 20 30**
- B. **20 40 60**
- C. **3**
- D. **2**
- E. **Compilation fails due to error at line 9.**

## Given

```
1. import java.util.stream.Collectors;
2. import java.util.stream.Stream;
3.
4. public class Whizlab{
5.
6.     public static void main(String[] args){
7.
8.         Stream<String> stream = Stream.of("12","13","3","1");
9.
10.        int avg = stream.collect(Collectors.averagingInt(s -> Integer.parseInt(s)));
11.        System.out.println(avg);
12.    }
13. }
```

## What is the output?

Please select :

- A. 7.25
- B. 7.0
- C. 7
- D. An Exception
- E. Compilation fails.

**Given**

```
1. import java.util.List;
2. import java.util.Map;
3. import java.util.stream.Collectors;
4. import java.util.stream.Stream;
5.
6. public class Whizlab{
7.     public static void main(String[] args){
8.
9.         Stream<Client> clients = Stream.of(new Client("Will","vps server",200),
10.             new Client("Rachel","java program",1200),new Client("Anthony","Configuration",1000),
11.             new Client("Rachel","PHP Script",900));
12.
13.     Map<String, List<Client>> groups = clients.collect(Collectors.groupingBy(Client::getName));
14.     System.out.println(groups);
15. }
16. }
17.
18. class Client{
19.
20.     double budget;
21.     String name,project;
22.
23.     Client(String n,String pr ,double p){
24.         budget = p;
25.         name = n;
26.         project = pr;
27.     }
28.
29.     public String getName(){ return name; }
30.
31.     public String toString(){ return name + "," + budget; }
32. }
```

**What is the output?**

Please select :

- A. {Anthony=[Anthony,1000.0], Rachel=[Rachel,1200.0, Rachel,900.0], Will=[Will,200.0]}
- B. {Anthony=[Anthony,1000.0], Rachel=[Rachel,1200.0, Will=[Will,200.0]]}
- C. {Anthony=[Anthony,1000.0], Rachel=[ Rachel,900.0], Will=[Will,200.0]}
- D. An exception is thrown.
- E. Compilation fails.

**Given**

```
1. // Assume all imports have done.
2. public class Whizlab{
3.     public static void main(String[] args){
4.
5.         List<Client> clients = new ArrayList<>();
6.         clients.add(new Client("Will","vps server",200));
7.         clients.add(new Client("Rachel","java program",1200));
8.         clients.add(new Client("Raj","PHP server",750));
9.
10.        Collections.sort(clients, Comparator.comparing(Client::getName));
11.        System.out.println(clients);
12.    }
13. }
14.
15. class Client{
16.
17.     double budget;
18.     String name,project;
19.
20.     Client(String n,String pr ,double p){
21.         budget = p;
22.         name = n;
23.         project = pr;
24.     }
25.
26.     public String getName(){
27.         return name;
28.     }
29.     public String toString(){
30.         return name + "," + budget;
31.     }
32. }
```

**What is the output?**

Please select :

- A. [Will,200.0, Rachel,1200.0, Raj,750.0]
- B. [Rachel,1200.0, Raj,750.0, Will,200.0]
- C. An exception is thrown.
- D. Compilation fails due to error at line 10.

**Given**

```
1. import java.util.ArrayList;
2. import java.util.Collections;
3. import java.util.List;
4. import java.util.function.BiFunction;
5.
6. public class Whizlab{
7.     public static void main(String[] args){
8.
9.         List<String> strings = new ArrayList<>();
10.        strings.add("Shane");
11.        strings.add("Rachel");
12.        strings.add("Raj");
13.
14.        BiFunction<String, String, Integer> func = (s1, s2) -> Integer.compare(s1.length(), s2.length());
15.
16.        Collections.sort(strings, func::apply);
17.
18.        System.out.println(strings);
19.    }
20. }
```

**What is the output?**

Please select :

- A. [Raj, Shane, Rachel]
- B. [Shane, Rachel, Raj]
- C. An Exception is thrown.
- D. Compilation fails due to error at line 14.
- E. Compilation fails due to error at line 18.

**Given**

```
1. import java.util.stream.IntStream;  
2.  
3. public class Whizlab{  
4.  
5.     public static void main(String[] args){  
6.         IntStream ints = IntStream.of(3,2,1,5,3,4,5);  
7.         System.out.print(ints.filter(e -> e>=3).distinct().average());  
8.     }  
9. }
```

**What is the output?**

Please select :

- A. **OptionalDouble[4.0]**
- B. **OptionalDouble[2.0]**
- C. **4**
- D. **3.28**
- E. **Compilation fails due to error at line 8.**

**Given**

```
1. import java.util.stream.IntStream;
2. public class Whizlab{
3.     public static void main(String[] args){
4.         IntStream ints = IntStream.of(2, 4, 6);
5.         System.out.println(ints.filter(x -> x<0).findFirst());
6.     }
7. }
```

**What is the output?**

Please select :

- A. **OptionalInt.empty**
- B. **-1**
- C. **null**
- D. **An Exception.**
- E. **Compilation fails.**

**Given**

```
1. import java.util.stream.IntStream;
2.
3. public class Whiz{
4.     public static void main(String[] args){
5.         IntStream ints = IntStream.of(3,2,1,5,3);
6.         ints.peek(System.out::print).allMatch(i -> i>1);
7.     }
8. }
```

**What is the output?**

Please select :

- A. **32153**
- B. **321**
- C. **No output.**
- D. **An Exception.**
- E. **Compilation fails.**

**Which of the following method of the Optional class can be used to get the value wrapped in optional?**

Please select :

- A. **orElseGet(T)**
- B. **value()**
- C. **orElse(Supplier)**
- D. **get()**
- E. **None of above.**

**Given**

```
1. import java.util.stream.DoubleStream;
2.
3. public class Whizlab{
4.     public static void main(String[] args){
5.         DoubleStream dbs = DoubleStream.of(1.1,1.0,0.8,0.6,1.5);
6.         dbs.skip(2).peek(in -> System.out.print((int)in)).limit(2).allMatch(d -> d<1);
7.     }
8. }
```

**What is the output?**

Please select :

- A. 11001
- B. 1100
- C. 00
- D. 11
- E. **Compilation fails.**

**Which of the following is not a valid primitive version of the Optional?**

Please select :

- A. **OptionalDouble**
- B. **OptionalInt**
- C. **OptionalLong**
- D. **OptionalFloat**
- E. **All given are valid.**

**Given**

```
1. import java.util.Optional;
2. import java.util.stream.Stream;
3.
4. public class Whizlab{
5.     public static void main(String[] args){
6.         Stream<String> streams = Stream.of("one","two","three");
7.         Optional<String> op =
8.             streams.filter(s -> s.length()>5).findFirst().flatMap(s->Optional.of("4"));
9.         System.out.println(op);
10.    }
11. }
```

**What is the output?**

Please select :

- A. **Optional.empty**
- B. **Optional[three]**
- C. **Optional[4]**
- D. **An exception.**
- E. **Compilation fails.**

**Given :**

```
1. import java.util.Optional;  
2.  
3. public class Whizlab{  
4.     public static void main(String[] args){  
5.         Optional<Integer> ops = Optional.of(8);  
6.         int i = ops.filter(p -> p>8).ifPresent(System.out::print).orElse(0);  
7.         System.out.println(i);  
8.     }  
9. }
```

**What is the output?**

Please select :

- A. 0
- B. 8
- C. 80
- D. An exception.
- E. Compilation fails.

**Given**

```
1. public class Whizlab{
2.
3.     public static void main(String[] args){
4.
5.         Stream<Trade> list = Stream.of(new Trade(1,1000,"John"),new Trade(5,200,"Alice"),
6.                                         new Trade(4,250,"Lisa"));
7.
8.         //insert here
9.         }
10.    }
11.
12.    class Trade{
13.        int quantity;;
14.        double price;
15.        double total;
16.        String buyer;
17.
18.        Trade(int q,double p,String b){
19.            quantity = q;
20.            price = p;
21.            total = p*q;
22.            buyer = b;
23.        }
24.
25.        public double getTotal(){
26.            return total;
27.        }
28.
29.        public String toString(){
30.            return buyer + ":" + total;
31.        }
32.    }
```

Which insert at line 8 will create Stream object for totals from the list stream?

Please select :

- A. DoubleStream totals = list.map(Trade::getTotal);
- B. Stream<Double> totals = list.mapToDouble(Trade::getTotal);
- C. Stream<Double> totals = list.map();
- D. DoubleStream totals = list.apply(Trade::getTotal);
- E. Stream<Double> totals = list.map(Trade::getTotal);

**Given**

```
1. import java.util.function.DoubleFunction;
2. import java.util.stream.DoubleStream;
3. import java.util.stream.Stream;
4.
5. public class Whizlab{
6.
7.     public static void main(String[] args){
8.
9.         Stream<Client> clients = Stream.of(new Client("Will","vps server",200),
10.                                         new Client("Rachel","java program",1200),new Client("Anthony","Configuration",1000));
11.
12.         DoubleFunction<Client> func = Client :: getBudget;
13.         DoubleStream prices = clients.mapToDouble( func);
14.         prices.forEachOrdered(d -> System.out.print(d + " "));
15.
16.     }
17.
18.     class Client{
19.
20.         double budget;
21.         String name,project;
22.
23.         Client(String n,String pr ,double p){
24.             budget = p;
25.             name = n;
26.             project = pr;
27.         }
28.
29.         public double getBudget(){ return budget; }
30.
31.         public String getName(){ return name; }
32.
33.         public String toString(){
34.             return name + "," + budget;
35.         }
36.     }
37.
```

**What is the output?**

Please select :

- A. 200.0 1200.0 1000.0
- B. 200.0 1000.0 1200.0
- C. Compilation fails due to error at line 12.
- D. Compilation fails due to error at line 13.
- E. Compilation fails due to multiple errors.

**Given**

```
1. import java.util.function.BinaryOperator;  
2.  
3. public class Whizlab{  
4.     public static void main(String [ ] args){  
5.         BinaryOperator<Integer> co = BinaryOperator.maxBy(Integer::compare);  
6.         System.out.println(co.apply(5, 4));  
7.     }  
8. }
```

**What is the output?**

Please select :

- A. 5
- B. 4
- C. 1
- D. -1
- E. Compilation fails.

### Given

```
1. import java.util.function.BiFunction;
2. import java.util.function.Function;
3.
4. public class Whizlab{
5.     public static void main(String [ ] args){
6.         BiFunction<Integer,Double,String> bf = (i,d) -> i.doubleValue() + d+ "";
7.         Function<String, Double> fun = Double::parseDouble;
8.         double d = bf.compose(fun).apply(2, 3.2);
9.         System.out.println(d);
10.    }
11. }
```

### What is the output?

Please select :

- A. 5
- B. 3.2
- C. 5.2
- D. An Exception.
- E. Compilation fails.

**Which of the following can be used to fill the blank?**

*BinaryOperator<Integer> operator = \_\_\_\_\_*

**Select 2 options.**

Please select :

- A. `i -> i/2;`
- B. `(x,y) -> System.out.print(x-y);`
- C. `(Integer x,Integer y) -> y +=x;`
- D. `Integer::rotateLeft;`
- E. **None of above.**

Given

```
1. import java.util.function.ObjIntConsumer;
2. import java.util.function.UnaryOperator;
3. public class Whizlab{
4.     public static void main(String [ ] args){
5.         UnaryOperator<Double> up = d -> d+2;
6.         UnaryOperator<Double> up2 = d -> d+3;
7.         UnaryOperator<Double> up3 = up.compose(up2);
8.         System.out.println(up3.apply(2.1));
9.     }
10. }
```

What is the output?

Please select :

- A. 4.1
- B. 5.1
- C. 7.1
- D. 2.1
- E. Compilation fails.

**Given**

```
1. import java.util.function.Function;  
2.  
3. public class Whizlab{  
4.     public static void main(String [ ] args){  
5.         Function<String, Double> fun = Double::parseDouble;  
6.         Double d = fun.compose((String s) -> s + ".01").andThen(s -> s*2).apply("1");  
7.         System.out.println(d);  
8.     }  
9. }
```

**What is the output?**

Please select :

- A. **1.01**
- B. **2.01**
- C. **2.02**
- D. **1.02**
- E. **Compilation fails.**

**Given**

```
1. import java.util.function.Function;  
2.  
3. public class Whizlab{  
4.     public static void main(String [ ] args){  
5.         Function<Integer, String> fun = s -> s.toString();  
6.         _____ fun2 = fun.andThen((String s) -> s+"2").compose((String s) -> Integer.parseInt(s));  
7.     }  
8. }
```

**Which of the following can be used to fill the blank?**

Please select :

- A. **Function<String, String>**
- B. **Function<String, Integer>**
- C. **Function<Integer, String>**
- D. **Function<Integer, Integer>**
- E. **None of above.**

**Given**

```
1. import java.util.function.Predicate;
2.
3. public class Whizlab{
4.     public static void main(String [ ] args){
5.         String in = " Whizlab ";
6.         Predicate<String> p = s -> s.indexOf("i") > 0;
7.         System.out.print(p.and(s -> s.length()>4).negate().test(in));
8.         System.out.print(p.negate().or(s -> s.length()>4).test(in));
9.     }
10. }
```

**What is the output?**

Please select :

- A. **falsefalse**
- B. **truefalse**
- C. **falsetrue**
- D. **An Exception is thrown.**
- E. **Compilation fails**

**Which of the following can be considered as functional interface but not includes in the function package?**

Please select :

- A. **Supplier**
- B. **Set**
- C. **Comparator**
- D. **Map**
- E. **Consumer**

**Which of the following is a valid statement?**

Please select :

- A. **ObjIntConsumer<Integer,int> cons = (l,i) -> System.out.println(i+l);**
- B. **ObjIntConsumer<int,Integer> cons = (l,i) -> System.out.println(i+l);**
- C. **ObjIntConsumer<Integer> cons = (l,i) -> System.out.println(i+l);**
- D. **ObjIntConsumer cons = (Integer l,int i) -> System.out.println(i+l);**
- E. **None of above.**

### Given

```
1. import java.util.*;
2.
3. class Df{
4.
5.     public static void main(String args[]){
6.         Locale loc = new Locale.Builder().setLanguage("en"). build();
7.
8.         System.out.print(loc.getDisplayLanguage(new Locale("En")));
9.         System.out.print(" " + loc.getDisplayCountry(new Locale("En")));
10.    }
11. }
```

Which can be true regarding above code? (Choose 2)

Please select :

- A. Compilation succeeds.
- B. Compilation fails.
- C. At line 6, we have done a mistake by not setting the region using setRegion() method.
- D. The output will be “English United States”
- E. The output will be “English”

**If you are asked to create a Locale object. The country should be India (region code – “IN”) and Language should be French. Which of the following ways you will choose? (Choose 2)**

Please select :

- A. **Locale loc = new Locale("in" , "FR");**
- B. **Locale loc = new Locale("IN" , "fr");**
- C. **Locale loc = new Locale("fr" , "IN");**
- D. **Locale loc = new Locale.Builder().setLanguage("fr").setRegion("IN").build();**
- E. **Locale loc = new Locale.Builder().setLanguage("fr"). build();**

## Given

```
1. import java.util.*;
2.
3. class Df{
4.
5.     public static void main(String args[]){
6.         Locale loc = new Locale.Builder().setRegion("CA").build(); //CA for Canada
7.
8.         System.out.print(loc.getDisplayLanguage(new Locale("En")));
9.         System.out.print( loc.getDisplayCountry(new Locale("En")));
10.    }
11. }
```

Which can be true regarding above code? (Choose 2)

Please select :

- A. Compilation succeeds.
- B. Compilation fails.
- C. At line 6, we have done a mistake by not setting the region using setLanguage() method.
- D. The output will be "English Canada"
- E. The output will be "Canada"

Consider following two classes.

```
import java.util.*;
public class SRBundel_de_CH extends
ListResourceBundle{
protected Object[][] getContents(){
    Object[][] resources = new
Object[1][2];
    resources[0][0] = "Hello";
    resources[0][1] = "Guten Tag";
    return resources;
}
}
```

```
import java.util.*;
public class SRBundel_hi_IN extends
ListResourceBundle{
protected Object[][] getContents(){
    Object[][] resources = new Object[1]
[2];
    resources[0][0] = "Hello";
    resources[0][1] = "Namaste";
    return resources;
}
}
```

```
4.
5.     Locale.setDefault(new Locale("IN"));
6.     ResourceBundle rb = ResourceBundle.getBundle("SRBundel", Locale.getDefault());
7.     System.out.print(rb.getString("Hello"));
8. }
9. }
```

What will be the result after the compiling and executing following code?

1. import java.util.\*;
2. public class RB{
3. public static void main(String[] a){

Please select :

- A. Guten Tag
- B. Namaste.
- C. A MissingResourceException will be thrown at the runtime.
- D. Compilation fails due to error at line 5.
- E. Compilation fails due to error at line 6.

**Consider following three statements.**

- I. Using the ResourceBundle.Control we can specify how to locate the ResourceBundle.
- II. It defines a set of callback methods that are invoked by the ResourceBundle.getBundle factory methods during the bundle loading process.
- III. Using the ResourceBundle.Control we can specify how long the resource bundle can exist in the cache.

**Which of the above statements are true?**

Please select :

- A. Only I.
- B. Only I and II.
- C. Only I and III.
- D. All.
- E. None.

**Given**

```
1. import java.util.*;
2.
3. class Whiz{
4.
5.     public static void main(String args[]){
6.         Locale loc = new Locale.Builder().setLanguage("en").build();
7.
8.         System.out.print(loc.getDisplayLanguage(new Locale("En")));
9.         System.out.print(" "+ loc.getDisplayCountry(new Locale("En")));
10.    }
11. }
```

**Which can be true regarding above code? (Choose 2)**

Please select :

- A. Compilation succeeds.
- B. Compilation fails.
- C. At line 6, we have done a mistake by not setting the region using setRegion() method.
- D. The output will be "English United States"
- E. The output will be "English"

. Given

```
Instant instant = Instant.now();
```

Which of the following will add 1 hours and 10 minutes to above instant?

Please select :

- A. **instant.plusDuration(Duration.ofHours(1).plusMinutes(10));**
- B. **instant.plus(Duration.of(1,10));**
- C. **instant.add(new Duration(1,10));**
- D. **instant.add(70, TemporalUnit.MINUTES);**
- E. **instant.plus(1, ChronoUnit.HOURS).plus(10, ChronoUnit.MINUTES);**

**The Instant class is locate in \_\_\_\_\_ package.**

Please select :

- A. **java.util**
- B. **java.time**
- C. **java.util.time**
- D. **java.lang.time**
- E. **java.sql**

### Given

```
1. import java.time.Duration;  
2. import java.time.LocalTime;  
3.  
4. public class Whizlab{  
5.     public static void main(String[] args) {  
6.         LocalTime s = LocalTime.of(10,10);  
7.         LocalTime e = LocalTime.of(20,1);  
8.         long due = Duration.between(s, e).toHours();  
9.         System.out.println(due);  
10.    }  
11. }
```

### What is the output?

Please select :

- A. 10
- B. -10
- C. 9
- D. An Exception.
- E. Compilation fails.

## Given

```
1. import java.time.MonthDay;  
2. import java.time.Year;  
3. import java.time.YearMonth;  
4.  
5. public class Whizlab{  
6.     public static void main(String[] args){  
7.         Year y = Year.of(2000);  
8.         YearMonth ym = y.atMonthDay(MonthDay.of(11,22));  
9.         System.out.println(ym);  
10.    }  
11.}
```

## What is the output?

Please select :

- A. 2000-11-22
- B. 2000-22-11
- C. 2000
- D. An Exception is thrown.
- E. Compilation fails.

## Given

```
1. import java.time.LocalDate;
2. import java.time.format.DateTimeFormatter;
3.
4. public class Whiz{
5.     public static void main(String[] args) {
6.         DateTimeFormatter format = DateTimeFormatter.ofPattern("y D");
7.         LocalDate ld = LocalDate.parse("2015 10",format);
8.         System.out.println(ld);
9.     }
10. }
```

## What is the output?

Please select :

- A. 2015-10
- B. 2015-10-01
- C. 2015-01-10
- D. An Exception is thrown.
- E. Compilation fails.

### Given

```
1. import java.time.LocalDate;
2.
3. public class Whizlab{
4.     public static void main(String[] args) {
5.         LocalDate l = LocalDate.of(2014, 1, 31).plusMonths(1);
6.         System.out.println(l);
7.     }
8. }
```

### What is the result?

Please select :

- A. 2014-01-31
- B. 2014-03-01
- C. 2014-02-28
- D. An Exception.
- E. Compilation fails. f

**Which of the following create ZoneId for following zone?**

**Africa/Harare**

Please select :

- A. **ZonedDateTime zone = ZoneId.get("Africa/Harare");**
- B. **ZonedDateTime zone = new ZoneId("Africa/Harare");**
- C. **ZonedDateTime zone = ZoneId.systemDefault("Africa/Harare");**
- D. **ZonedDateTime zone = ZoneId.apply("Africa/Harare");**
- E. **ZonedDateTime zone = ZoneId.of("Africa/Harare");**

**Which of the following can be used to fill the blank?**

**BiConsumer<Integer, String> consumer = \_\_\_\_\_**

Please select :

- A. **(i,s) -> System.out.println(i+s)**
- B. **(i,s) -> i+s.length()**
- C. **Integer::toString**
- D. **(Integer i, s) -> System.out.println(i+s)**
- E. **None of above.**