MOCK_TEST_DAY6_Excep_V2

* Required

1

Participant's Name *

Pavithra J

2

Email ID *

j.pavithra@ltimindtree.com

3

Which of the following types are not recommended to catch in your application? (Choose two.) * (1 Point)

Please select 2 options.

- [ ] A. Exception
- [ ] B. CheckedException
- [x] C. Throwable
- [ ] D. RuntimeException
- [ ] E. UncheckedException
- [x] F. Error

4

How many of these custom exceptions are unchecked exceptions?

class ColoringException **extends IOException {}**
class CursiveException extends WritingException {}
class DrawingException extends IllegalStateException {}
class SketchingException extends DrawingException {}
class WritingException extends **Exception {}** * (1 Point)

○ A. None

○ B. One

● C. Two

○ D. Three

○ E. Four

○ F. Five

5

Which expressions, when inserted into the blank in the following class, allow the code to compile? (Choose two.)

package sun;
import java.io.*;
public class Beach {
class TideException extends Exception {}
public void surf() throws RuntimeException {
 try {
 throw new TideException();
 } catch (_____) {}
 }
} * (1 Point)

Please select 2 options.

☐ A. Exception a | RuntimeException f

☐ B. IllegalStateException | TideException t

☐ C. TideException | IOException i

☑ D. TideException | Exception x

☐ E. Error e

☑ F. Exception z

What is the output of the following application?

```
package office;
import java.io.*;
public class Printer {
public void print() {
 try {
 throw new FileNotFoundException();
 } catch (Exception | RuntimeException e) {
 System.out.print("Z");
 } catch (Throwable f) {
 System.out.print("X");
 } finally {
 System.out.print("Y");
 }
}
public static void main(String... ink) {
 new Printer().print();
}
} * (1 Point)
```

○ A. Y

○ B. XY

○ C. ZY

● D. The code does not compile.

○ E. The code compiles, but a stack trace is printed at runtime.

○ F. None of the above.

7

Given

What is the output? * (1 Point)

```
public class Test{
    private int num = 1;
    private int div = 0;

    public void divide() {
        try {
            num = num / div;
            System.out.print("Exception");
        }
        catch (ArithmeticException ae) { num = 100; }
        catch (Exception e) { num = 200; }
        finally { num = 300; }
        System.out.print(num);
    }
    public static void main(String args[])
    {
        Test test = new Test();
        test.divide();
    }
}
```

● A. 300

○ B. Exception

○ C. 200

○ D. 100

8

What is the output of the following application?

```
 public class Football {
public static void main(String officials[]) {
try {
System.out.print('A');
throw new RuntimeException("Out of bounds!");
} catch (ArrayIndexOutOfBoundsException t) {
System.out.print('B');
throw t;
} finally {
System.out.print('C');
}
}
} *  (1 Point)
```

○ A. ABC

○ B. ABC, followed by a stack trace for a RuntimeException

● C. AC, followed by a stack trace for a RuntimeException

○ D. None of the above

9

What will be the result of compiling and executing Test class?

```
 class MyResource implements AutoCloseable {
@Override
public void close() {
System.out.println("Closing");
}
}
public class Test {
 public static void main(String[] args) {
 try(AutoCloseable resource = new MyResource()) {
}
 }
 }
```

What will be the result of compiling and executing Test class? * (1 Point)

◉ A. Compilation error in Test

◯ B. Compilation error in MyResource

◯ C. Runtime error

◯ D. Closing

10

Given:

and omitting the throws FooException clause results in a compilation error.
 Which statement is true about FooException? * (1 Point)

```
public method foo() throws FooException {

    ...

}
```

◯ A. FooException is a subclass of RuntimeError.

◯ B. FooException is unchecked.

◯ C. The body of foo can only throw FooException.

◉ D. The body of foo can throw FooException or one of its subclasses.

11

Given

What must be added in line 1 to compile this class? * (1 Point)

```
import java.io.FileNotFoundException;
import java.io.IOException;

public class Tester {
    public static void main(String[] args) {
        try {
            doA();
        } //line 1
    }
    private static void doA() throws IOException, IndexOutOfBoundsException {
        if (false) {
            throw new FileNotFoundException();
        } else {
            throw new IndexOutOfBoundsException();
        }
    }
}
```

○ A. catch(IOException e) { }

○ B. catch(FileNotFoundException | IndexOutOfBoundsException e) { }

○ C. catch(FileNotFoundException | IOException e) { }

○ D. catch(IndexOutOfBoundsException e) { }catch(FileNotFoundException e) { }

○ E. catch(FileNotFoundException e) { }catch(IndexOutOfBoundsException e) { }

12

Which of the following can be inserted into Lion to make this code compile? (Choose all that apply.)

class HasSoreThroatException extends Exception {}
class TiredException extends RuntimeException {}
interface Roar {
void roar() throws HasSoreThroatException;
}
class Lion implements Roar {
// INSERT CODE HERE
} * (1 Point)

Please select 4 options.

☑ 1. public void roar() {}

☐ 2. public int roar() throws RuntimeException {}

☐ 3. public void roar() throws Exception {}

☑ 4. public void roar() throws HasSoreThroatException {}

☑ 5. public void roar() throws IllegalArgumentException {}

✓ 6. public void roar() throws TiredException {}

13

What is printed by the following program?

* (1 Point)

```
1:  public class DoSomething {
2:      public void go() {
3:          System.out.print("A");
4:          try {
5:              stop();
6:          } catch (ArithmeticException e) {
7:              System.out.print("B");
8:          } finally {
9:              System.out.print("C");
10:         }
11:         System.out.print("D");
12:     }
13:     public void stop() {
14:         System.out.print("E");
15:         Object x = null;
16:         x.toString();
17:         System.out.print("F");
18:     }
19:     public static void main(String n[]) {
20:         new DoSomething().go();
21:     }
22: }
```

○ a. AE

○ b. AEBCD

○ c. AEC

○ d. AECD

○ e. AE followed by a stack trace

○ f. AEBCD followed by a stack trace

◉ g. AEC followed by a stack trace

○ h. A stack trace with no other output

14

Which two statements are correct about try blocks? (Choose two.) * (1 Point)

Please select 2 options.

☑ A. A try block can have more than one catch block.

☐ B. A finally block in a try-with-resources statement executes before the resources declared are closed.

☑ C. A finally block must be immediately placed after the try or catch blocks.

☐ D. A try block must have a catch block and a finally block.

Given:

public class Foo () throws FooException {

    . . .

}
  and omitting the throws FooException clause results in a compilation error. Which statement is true about FooException? " * (1 Point)

◯ A. FooException is a subclass of RuntimeError.

◯ B. FooException is unchecked.

◯ C. The body of foo can only throw FooException.

⦿ D. The body of foo can throw FooException or one of its subclasses.

Given code of Test.java file:

1. public class Test {

2.   public static void main(String[] args) {

3.   try {

4.   try {

5.   System.out.println(args[1]); //Line n1

6.   } catch(RuntimeException e) {

7.   System.out.print("INHALE-"); //Line n2

8.   throw e; //Line n3

9.   } finally {

10.   System.out.print("EXHALE-"); //Line n4

11.   }

12.   } catch(RuntimeException e) {

13.   System.out.print("INHALE-"); //Line n5

14.   } finally {

15.   System.out.print("EXHALE"); //Line n6

16.   }

17. }

18. }

What will be the result of compiling and executing Test class?
  * (1 Point)

○ A. INHALE-EXHALE-

● B. INHALE-EXHALE-INHALE-EXHALE

○ C. INHALE-EXHALE

○ D. INHALE-EXHALE-INHALE

17

Given:

```
public class Test {
class L extends Exception { }
class M extends L { }
class N extends RuntimeException { }
public void p() throws L { throw new M(); }
public void q() throws N { throw new N(); }
public static void main(String[] args) {
try {
Test t = new Test();
t.p();
t.q():
} /*line 1*/ {
System.out.println(""Exception caught"");
}
}
}
```

What change on line 1 will make this code compile? * (1 Point)

● A. Add catch (L |N e).

○ B. Add catch (L |M N e).

○ C. Add catch (L e).

○ D. Add catch (N | L | M e).

○ E. Add catch (M |L e).

Never give out your password. [Report abuse](#)