

EXCEPTION

1. Apa yang akan dicetak pada saat program ini dicetak dan dijalankan?

```
package jenisoal;

public class CobaException {
    public static void main(String args[]) {
        int i=1, j=1;
        try {
            i++;
            j--;
            if(1/j > 1)
                i++;
        }
        catch(ArithmeticException e) {
            System.out.println(0);
        }
        catch(ArrayIndexOutOfBoundsException e) {
            System.out.println(1);
        }
        catch(Exception e) {
            System.out.println(2);
        }
        finally {
            System.out.println(3);
        }
        System.out.println(4);
    }
}
```

0

1

2

3

4

2. Pada kondisi bagaimana klausa finally tidak akan dieksekusi?

Kode didalam block try memiliki statement return

Kode didalam block try telah dieksekusi dengan benar

Method `System.exit()` dipanggil dalam block try atau catch

Kode didalam block try memiliki statement break

pembuatan checked exception handler?

- Subclass class dari class Throwable .
- Subclass class dari class Exception .
- Subclass class dari class Error .
- Subclass class dari class RuntimeException .
- Tidak ada satupun dari pilihan di atas yang benar karena checked exception tidak bisa ditulis

4. Mana dari potongan kode berikut ini yang merupakan cara terbaik untuk melempar sebuah exception? Asumsikan bahwa semua variable yang belum dideklarasikan, telah dideklarasikan di tempat yang lain yang masih bisa diakses ?

```
1. Exception e = new IOException("File tidak ditemukan");
2. if (!f.exists()){ // f adalah sebuah object file
3.     throw e;
4. }
```

```
1. if (!f.exists()) { // f adalah object dari File
2. throw new IOException("File" + f.getName()); +"tidak
ditemukan");
3. }
```

```
1. if (!f.exists()){
2. throw IOException;
3. }
```

```
1. if (!f.exists()){
2. throw "File tidak ditemukan";
3. }
```

```
1. if(!f.exists()) { // f adalah object File
2. throw new IOException();
3. }
```

5. Apa yang harus dilakukan oleh programmer untuk menangani user-defined exception?

- Menuliskan exception sebagai start method
- Menangani exception sebagai sebuah error
- Menuliskan exception sebagai bagian dari class Component
- Tidak ada perbedaan antara exception tersebut dengan exception yang lain

6. Dari pernyataan-pernyataan dibawah ini, mana yang mendeskripsikan secara benar mengenai langkah yang harus diambil oleh programmer sehingga kode yang terdiri dari pernyataan-pernyataan dapat melempar sebuah exception?

Membiarkan sistem operasi menangani semua exception

Mendeklarasikan exception dalam klausa throws, atau menempatkan pernyataan tersebut dalam block try/catch/finally

Menambahkan kode dari class handler untuk menangkap exception

Mengabaikan semua exception

7. Mana yang merupakan definisi yang benar dari sebuah method m, yang diasumsikan bahwa method tersebut throws IOException dan memiliki nilai kembalian void. Asumsikan juga bahwa method tersebut tidak menerima argument apapun.

void m() throws IOException{ }

void m() throw IOException{ }

void m(void) throws IOException{ }

m() throws IOException{ }

void m() { } throws IOException

8. Package mana yang memiliki class IOException?

java.util

java.text

java.error

java.io

9. Perhatikan kode berikut ini:

```
1. public class Assertification{
2. public static void main(String[] args ){
3. asswer args.length == 0;
4. }
5. }
```

Kondisi manakah yang benar diantara pilihan jawaban dibawah ini, jika kode diatas dapat melewati AssertionError?

- Aplikasi diatas harus dijalankan dengan -enableassertions flag atau assertion-enabling flag yang lain
- Array args harus memiliki 0 elements
- Array args harus memiliki satu element atau lebih

10. Method hallo() bisa melempar java.io.IOException, java.lang.RuntimeException, atau java.net.MalformedURLException (yang merupakan subclass dari java.io.IOException). Import yang dibutuhkan oleh class-class berikut ini telah dideklarasikan untuk exception-exception tersebut diatas. Mana diantara pilihan-pilihan dibawah ini yang benar? (Pilih semua jawaban yang menurut Anda benar)

- ```
1. public class ClassSaya {
2. public void aMethod() {
3. hallo();
4. }
5. }
```

SALAH

- ```
1. public class ClassSaya{
2. public void aMethod() throws
3. IOException {
4. hallo();
5. }
6. }
```

- ```
1. public class ClassSaya {
2. public void aMethod throws
3. RuntimeException {
4. hallo();
5. }
6. }
```

- ```
1. public class ClassSaya{
2. public void aMethod(){
3. try{
4. hallo();
5. }
6. catch(IOException){
7. e. printStackTrace();
8. }
9. }
10.}
```

SALAH

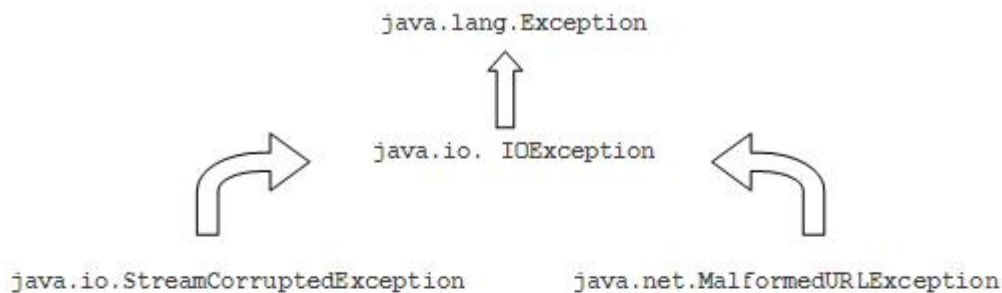
```

 1. public class ClassSaya{
2. public void aMethod()
3. throws MalformedURLException{
4. try{ hallo(); }
5. catch ( IOException e){
6. // abaikan disini
7. }
8. }
9. }
10.
11. public class ClassAnda
12. extends ClassSaya{
13. public void aMethod()
14. throws java.io.IOException {
15. super.aMethod();
16. }
17. }

```

SALAH

11. Perhatikan susunan diagram berikut ini :



Perhatikan juga potongan kode dibawah ini:

```

1. try {
2. // asumsikan bahwa s telah didefinisikan sebelumnya
3. URL u = new URL(s)
4. // in adalah sebuah ObjectInputStream
5. Object o = in.readObject()
6. System.out.println("Sukses");
7. }
8. catch(MalformedURLException e){
9. System.out.println("Kesalahan pada URL");
10. }
11. catch(StreamCorruptedException e){
12. System.out.println("File tidak lengkap");
13. }
14. catch (Exception e){
15. System.out.println("Kesalahan umum");
16. }
17. finally{
18. System.out.println("Mengerjakan finally");
19. }
20. System.out.println("Diluar block");

```

Baris mana yang akan menjadi output jika method pada baris 3 dan baris 5 sukses dijalankan tanpa melewati exception apapun?

Sukses

Kesalahan pada URL

File tidak lengkap

Mengerjakan finally

Diluar block

12. Block atau baris kode mana yang akan dieksekusi apabila block try telah terpenuhi tanpa terjadi exception?

main

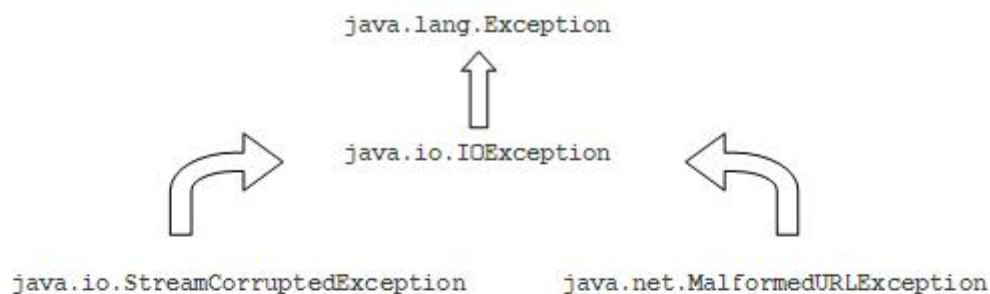
Catch salah

Finally

static

return

13. Perhatikan susunan diagram berikut ini :



Perhatikan juga potongan kode dibawah ini:

```
1. try {
2. // asumsikan bahwa s telah didefinisikan sebelumnya
3. URL u = new URL(s)
4. // in adalah sebuah ObjectInputStream
5. Object o = in.readObject()
6. System.out.println("Sukses");
7. }
8. catch(MalformedURLException e){
9. System.out.println("Kesalahan pada URL");
10. }
11. catch(StreamCorruptedException e){
12. System.out.println("File tidak lengkap");
13. }
14. catch (Exception e){
15. System.out.println("Kesalahan umum");
16. }
17. finally{
18. System.out.println("Mengerjakan finally");
19. }
20. System.out.println("Diluar block");
```

Baris mana yang akan menjadi output apabila konstruktor pada baris 3 throws MalformedURLException?

Sukses

- File tidak lengkap
- Kesalahan umum
- Mengerjakan finally
- Diluar block

14. i pernyataan-pernyataan berikut ini, pilih semua jawaban yang menurut Anda benar?

Pada setiap block try-catch, harus ada minimal sebuah block catch

Sebuah block try dapat diikuti oleh beberapa block finally salah

Sebuah block try harus diikuti oleh minimal sebuah block finally atau catch

Jika block catch dan finally telah didefinisikan, block catch harus didahului oleh block finally

15. Sebuah variabel dalam local method bisa diakses dari dalam block try-catch. Dimana variabel tersebut sebaiknya dideklarasikan?


Variabel tersebut seharusnya dideklarasikan local sebelum memasuki block try-catch

Variabel tersebut seharusnya bisa dideklarasikan didalam try block dan block catch

Variabel tersebut seharusnya dideklarasikan dalam block catch

Variabel tersebut seharusnya bisa dideklarasikan dalam block try

16. Apa yang akan dicetak apabila kode dibawah ini decompile dan dijalankan?



```
package jenidoal;

/**...*/
public class CobaException {
    public static void main(String args[]) {
        int i=1, j=1;
        try {
            i++;
            j--;
            if(i == j)
                i++;
        }
        catch(ArithmeticException e) {
            System.out.println(0);
        }
        catch(ArrayIndexOutOfBoundsException e) {
            System.out.println(1);
        }
        catch(Exception e) {
            System.out.println(2);
        }
        finally {
            System.out.println(3);
        }
        System.out.println(4);
    }
}
```

0

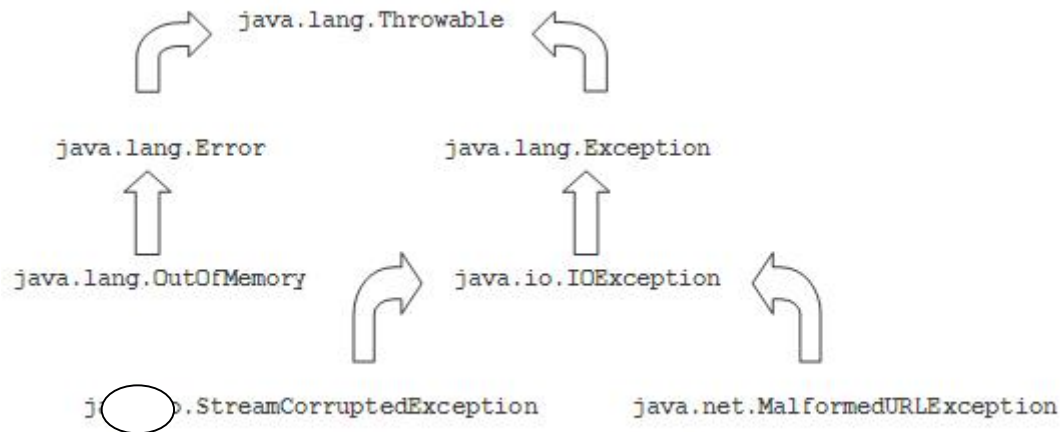
1

2

3

4

17. Perhatikan susunan diagram berikut ini:



```
1. try {
2. // asumsikan bahwa s telah didefinisikan sebelumnya
3. URL u = new URL(s)
4. // in adalah sebuah ObjectInputStream
5. Object o = in.readObject()
6. System.out.println("Sukses");
7. }
8. catch(MalformedURLException e){
9. System.out.println("Kesalahan pada URL");
10. }
11. catch(StreamCorruptedException e){
12. System.out.println("File tidak lengkap");
13. }
14. catch (Exception e){
15. System.out.println("Kesalahan umum");
16. }
17. finally{
18. System.out.println("Mengerjakan finally");
19. }
20. System.out.println("Diluar block");
```

Baris mana yang akan menjadi output apabila method pada baris ke 5 throws OutOfMemoryError?

- Sukses
- Kesalahan pada URL
- File tidak lengkap
- Kesalahan umum
- Mengerjakan finally
- Diluar block

Submit