Questions on Exception Handling

1) What is the exception handling?
Exception Handling is a way of responding to unwanted events that might occur in execution of Computer program.

2) What is the use of finally block?
"Finally" block is used to contain code that needs to execute regardless of whether an exception occurs or not.

3) what kind of information gives Exception message in console?
Exception message contains information about the type of exception that has occurred. E.g.

4) In case of something by zero which exception will arise?
"ArithmeticException"

5) What is the order Catch blocks?
Child classes followed by Parent classes

6) How can we handle exceptions?
We add the code that needs to be tested in "Try" Block and use "catch" block to catch any exceptions that needs to be handled.

7) Write any 3 Error classes?
IOError, AssertionError, ThreadDeath

8) Which one is the super class for RuntimeException?
Exception Class

9) Is it possible to write a try within a try block?
Yes, it is possible to write try block inside another try block

10) Is there any case when finally will not be executed?
Only if there is a condition to terminate the execution of the code using System.exit(0).

11) In which scenario we have to keep try inside another try?
If the execution of the code has a possibility of generating another exception.

12) The exception class is available in ____ package
Java.Lang

13) What is the NullPointerException?
NullPointerException occurs when code tries to access an object that is Null.

14) What is the difference between final, finally?
Final is an access modifier, Finally is block that is used to make sure a part of code whether an exception occurs in the code or not.

15) Give an example of ArrayIndexOutOfBoundsException?
Int[] a = [1,2,3]
System.out.println(a[4]);
This will give ArrayIndexOutOfBoundsException.

16) What is the difference between Exception and Error?
Errors are Unrecoverable system issues
Exceptions are unexpected events.

17) SQLException is a unchecked exception?(Yea **OR** No)
No, SQLException is an Checked excpeption.

18) NumberFormatException is unchecked Exception?(Yes **OR** No)
Yes, unchecked Exception

19) NullPointerException comes under Unchecked Exception?(True **OR** False)
False, NullPointerException is Checked Expception.

20) What is the use of ex.PrintStackTrace?
PrintStackTrace is used to get the stack memory of the object that has caused the exception.

21) Can we use local variables of main method inside try or catch block?
Yes, We can use local variables inside try or catch block.

22) Can we use local variable of try block inside a catch block?
Yes, we can use local variables inside catch block.

23) What is the super most class for Exception?
Object class.

24) Inside catch block any exception is occurred then program will get terminate?(True/False). Give reason.
True, the program will terminate if the exception is not caught.

25) Give an Example on an unreachable statement.
```
Try{}
Catch(Exception e){}
Finally{return 0;}
Return 0;
```

Last return statement is unreachable.

26)
```
class A {
public static void main(String args[]) {
try
{            int a, b;
             b = 0;
             a = 5 / b;
             System.out.print("A");
}
catch(ArithmeticException e)
{
System.out.print(e);
}
}
}
```

Output: java.lang.ArithmeticException

27)
```
class B {
public static void main(String args[]) {
try
{   int a[] = {1, 2,3 , 4, 5};
for (int i = 0; i < 7; ++i)
System.out.print(a[i]);
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.print("0");
}
}
```

```
}

Output:1
2
3
4
5
0
```

28)
```
class C  {
public static void main(String[] args)  {
try
{
 System.out.println(1);
int i=10/0;
System.out.println(2);
}
catch(ArithmeticException ex)
{
System.out.println(3);
System.out.println(ex.getMessage());
System.out.println(4);
}
System.out.println(5);
}
}

Output:
1
3
/ by zero
4
```

29)
```
class D   {
public static void main(String[] args) {
try
{3
int i;
return ;
}
catch (Exception e)
{
System.out.println("inCatchBlock");
}
finally
{
System.out.println("inFinallyBlock");
}
}
}

Output: inFinallyBlock
```

30)
```
class E  {
public static void main(String[] args) throws
ClassNotFoundException      {
System.out.println(1);
if (true)
{
throw new ClassNotFoundException();
}
System.out.println(2);
}
}

Output: 1
Exception in Thread “Main”
```

31)
```
public class F    {
public static void main(String[] args)    {
try
{
System.out.println(1);
int i=10/0;
System.out.println(2);
}
catch(NumberFormatException ex)
{
System.out.println(4);
System.out.println(ex.getMessage());
System.out.println(5);
}
finally
{
System.out.println(6);
}
System.out.println(5);
}
}

Output: 1
6
Exception in Main thread
```

32)
```
public class G {
public static void main(String[] args) {
try
{
System.out.println(1);
String s=null;
System.out.println(s);
System.out.println(s.length());
System.out.println(2);
}
catch(NullPointerException ex)
{
System.out.println(4);
System.out.println(5);
}
System.out.println(6);
}
}

Output: 1
Null
4
5
6
```

33)
```
public class   H  {
public static void main(String[] args) {
try
{
System.out.println(1);
String s=null;
System.out.println(s);
System.out.println(s.length());
System.out.println(2);
}
catch(NullPointerException ex)
{
System.out.println(4);
System.out.println(s);
System.out.println(5);
```

```
        }
                System.out.println(6);
        }
        }
```

Output: 1
Null
4
5
6

```
34)  class I  {
        int test1()  {
        try
        {
                //some stmts
        }
        catch (ArithmeticException ex)
        {
        }
        finally
        {
        }
        return 10;
        }
        int test2()  {
        try
        {
                //some stmts
        }
        catch (ArithmeticException ex)
        {
        }
        finally
        {
                return 30;
        }
        }
        }
```

Output:
10
30

```
35) public class J  {
        public static void main(String a[]) {
        try
        {
                for(int i=5;i>=0;i--){
                System.out.println(10/i);
        }
        }
        catch(Exception ex)
        {
        System.out.println("Exception Message:+ex.getMessage()");
                ex.printStackTrace();
        }
        System.out.println("After for loop...");
        }
        }
```

Output:
2
2
3
5
10
Exception Message:+ex.getMessage()
After for loop...

```
36) public class K   {
        public static void main(String[] a)  {
        try
        {
                int i = 10/0;
        }
        catch(Exception ex)
        {
                System.out.println("Inside 1st catch Block");
        }
        finally
        {
                System.out.println("Inside 1st finally block");
        }
        try
        {
                int i = 10/10;
        }
        catch(Exception ex)
        {
                System.out.println("Inside 2nd catch Block");
        }
        finally
        {
                System.out.println("Inside 2nd finally block");
        }
        }
        }
```

Output:
Inside 1st catch block
Inside 1st Finally block
Inside 2nd Finally block

```
37) class N {
        public static void main(String[] args)   {
        String languages[] = { "C", "C++", "Java", ".Net", "C#" };
        try
        {
                for (int i = 1; i <= 5; i++) {
                System.out.println(languages[i]);
        }
        }
        catch (Exception e)
        {
                System.out.println(e);
        }
        }
        }
```

Output:
C++
Java
.Net
C#
ArrayOutOfIndexException

```
38)  public class P  {
        static int test()  {
        try
        {
                return 10;
```

```
}
catch(NumberFormatException ex)
{
        return 20;
}
finally
{
        return 30;
}
return 40;
}
public static void main(String[] args) {
System.out.println(1);
System.out.println(test());
System.out.println(2);
}
}
```

Output:
Unreachable return statement error.

39)
```
public class Q  {
    static int test()  {
    try
    {
    }
    catch(NumberFormatException ex)
    {
            return 20;
    }
    finally
    {
    }
    return 40;
    }
    public static void main(String[] args)    {
    System.out.println(1);
    System.out.println(test());
    System.out.println(2);
    }
    }
```

Output:
1
40
2

40)
```
class R {
    public static int test()    {
    try
    {
            return 0;
    }
    finally
    {
            System.out.println("Inside Finally block");
    }
    }
    public static void main(String args[])  {
            System.out.println(R.test());
    }
    }
```

Output:
Inside Finally block
0

41)
```
class S   {
    public static void main(String[] args)   {
    try
    {
            long data[] = new long[1000000000];
    }
    catch (Exception e)
    {
            System.out.println(e);
    }
    finally
    {
        System.out.println("finally block will execute always.");
    }
    }
    }
```

Output:
Finally block will execute always
Java Heap memory Exception

42)
```
public class X   {
    public static void main(String [] args) {
    try
    {
            badMethod();
            System.out.print("A");
    }
     catch (RuntimeException ex)
    {
             System.out.print("B");
    }
    catch (Exception ex1)
    {
             System.out.print("C");
    }
    finally
     {
            System.out.print("D");
    }
            System.out.print("E");
    }
    public static void badMethod()
    {
            throw new RuntimeException();
    }
    }
```

Output:
B
D
E

43)
```
class Fex    {
    public static void main(String[] args) throws
    ClassNotFoundException    {
    System.out.println(1);
    if (true)
    {
            throw new ClassNotFoundException();
    }
            System.out.println(2);
    }
    }
```

Output: 1
ClassNotFoundException

44)
```
public class T  {
    public static void main(String[] args) {
    try
    {
        return;
    }
    finally
    {
        System.out.println( "Finally" );
    }
    }
    }
```

Output:
Finally

45)
```
public class Z {
    public static void main(String[] args) {
    try
    {
        System.out.println(1);
        int i=10/0;
    }
    catch(NullPointerException ex)
    {
        System.out.println(4);
    try
    {
        int i=23/0;
    }
    catch(ArithmeticException ex)
    {
        System.out.println(5);
    }
    finally
    {
        System.out.println(6);
    }
    System.out.println(7);
    }
    }
```

Output:
1
5
6
7
ArithmeticException in Main Thread

46)
```
public class K   {
    static int test()       {
        try
        {
        }
        catch(NumberFormatException ex)
        {
            return 20;
        }
        finally
        {
            return 30;
        }
        return 40;
    }
    public static void main(String[] args) {
    System.out.println(1);
    System.out.println(test());
    System.out.println(2);
    }
    }
```

Output:
Unreachable statement.

47)
```
public class F  {
    public static void main(String[] args) {
    try
    {
        System.out.println(1);
        String s=null;
        System.out.println(s.length());
    }
    catch(NullPointerException ex)
    {
        System.out.println(4);
    try
    {
        int i=23/0;
    }
        System.out.println(5);
    }
    finally
    {
        System.out.println(6);
    }
    System.out.println(7);
    }
    }
```

Output:
Try without catch error.

48)
```
public class Test {
    public static void main(String args[]) {
    try
    {
        int a[]=new int[5];
        a[5]=30/0;
    }
     catch(ArithmeticException e)
    {
        System.out.println("task1 is completed");
    }
    catch(ArrayIndexOutOfBoundsException e)
    {
        System.out.println("task 2 completed");
     }
    catch(Exception e)
     {
        System.out.println("common task completed");
     }
    System.out.println("rest of the code...");
    }
    }
```

Output:
Task1 is completed
Rest of the code

49)
```
class Test1 {
    public static void main(String args[]) {
    try
    {
        int a[]=new int[5];
        a[5]=30/0;
    }
    catch(Exception e)
    {
        System.out.println("common task completed");
     }
     catch(ArithmeticException e)
    {
        System.out.println("task1 is completed");
    }
    catch(ArrayIndexOutOfBoundsException e)
    {
          System.out.println("task 2 completed");
    }
    System.out.println("rest of the code...");
    }
    }
```

Output:
Exception already caught error.

50)
```
class Test2    {
    void m()
    {
        int i=50/0;
     }
    void n()
    {
         m();
    }
     void p()
    {
        try
        {
            n();
        }
        catch(Exception e)
        {
            System.out.println("exception handled");
        }
    }
    public static void main(String args[])  {
    Test2 obj=new Test2();
    obj.p();
    System.out.println("normal flow...");
    }
    }
```

Output:
Exception handled
Normal flow