Exception Class is a parent class for all the exception classes.

```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(ArithmeticException e)
    {
        System.out.println("catch Block executed");
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA aobj=new ClassA();
    aobj.meth1();
```

```
meth1() called
10
try block executed
Enter a number
5
====>4
finally Block executed
30
```

```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(ArithmeticException e)
    {
        System.out.println("catch Block executed");
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA aobj=new ClassA();
    aobj.meth1();
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
finally Block executed
30
```

```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(ArithmeticException e)
    {
        System.out.println("catch Block executed");
        System.out.println(e.getMessage()); // Reason of the Exception
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA aobj=new ClassA();
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
/ by zero
finally Block executed
30
```

```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(ArithmeticException e)
    {
        System.out.println("catch Block executed");
        //System.out.println(e.getMessage()); // Reason of the Exception
        System.out.println(e.toString());// Name & Reason of the Exception

    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
finally Block executed
30
```

```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(ArithmeticException e)
    {
        System.out.println("catch Block executed");
        //System.out.println(e.getMessage()); // Reason of the Exception
        //System.out.println(e.toString());// Name & Reason of the Excepti
        e.printStackTrace(); // Complete info about the Exception
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
```

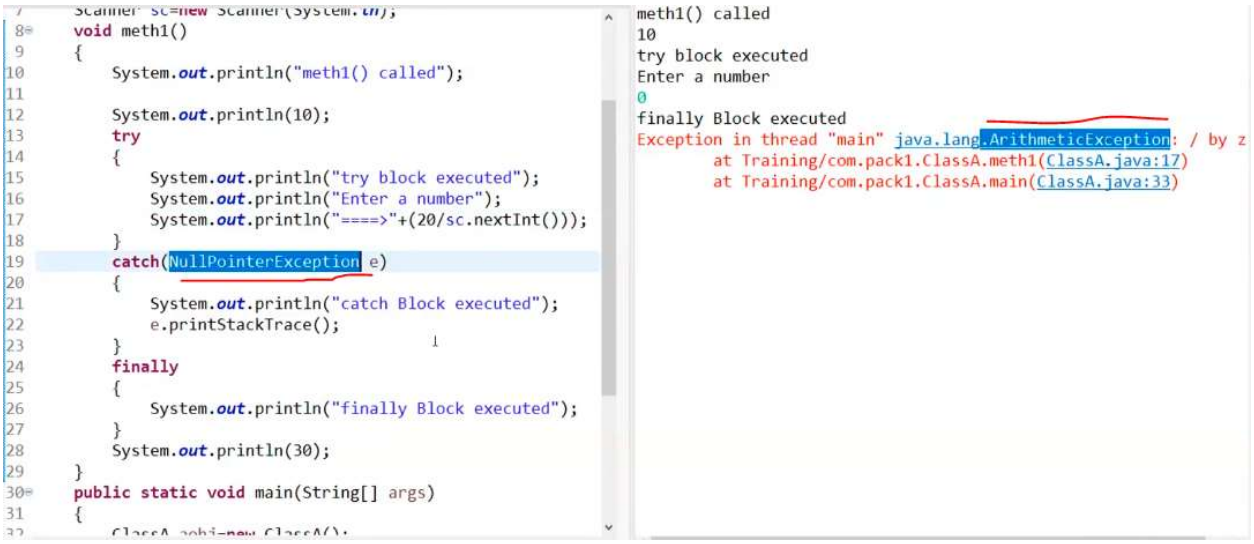
```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
        at Training/com.pack1.ClassA.meth1(Cla
        at Training/com.pack1.ClassA.main(Clas
finally Block executed
30
```

Methods To Display Exception Information

- Throwable class defines the following methods to print exception information to the console.

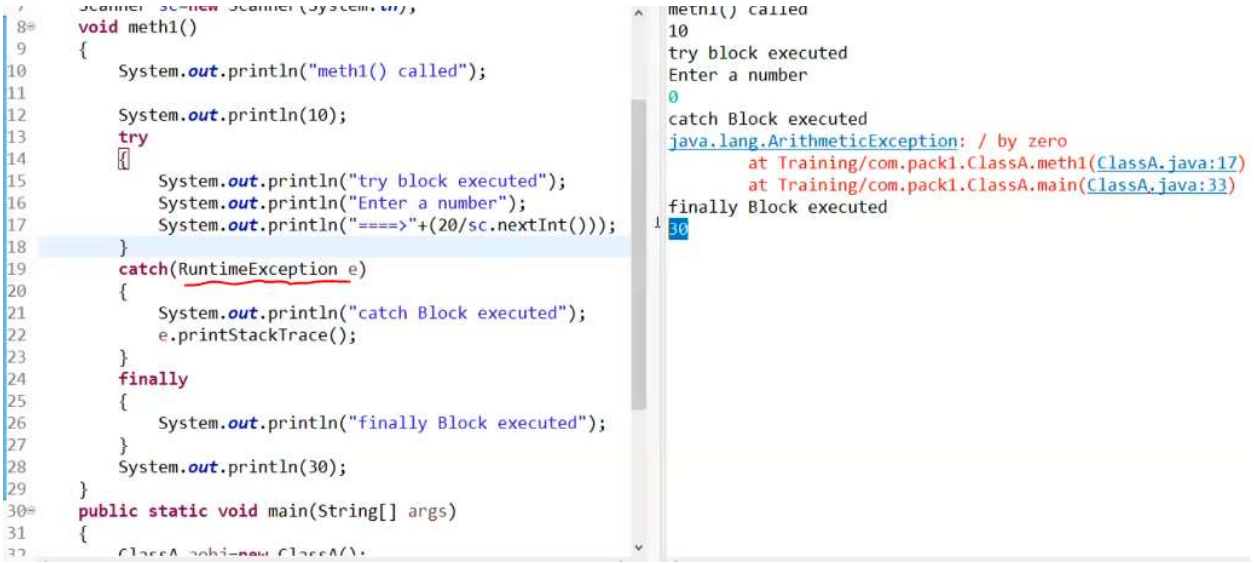
Method Name	Description
printStackTrace()	Name of the exception: description of exception Stack trace
toString()	Name of the exception: description of exception
getMessage()	Only Description

```java
import java.util.Scanner;

public class ClassA
{
    Scanner sc=new Scanner(System.in);
    void meth1()
    {
        System.out.println("meth1() called");

        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("====>"+(20/sc.nextInt()));
        }
        catch(ArithmeticException e)
        {
            System.out.println("catch Block executed");
            //System.out.println(e.getMessage()); // Reason of the Exception
            //System.out.println(e.toString());// Name & Reason of the Exception
            e.printStackTrace(); //******** Complete info about the Exception
        }
        finally
        {
            System.out.println("finally Block executed");
        }
        System.out.println(30);
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```



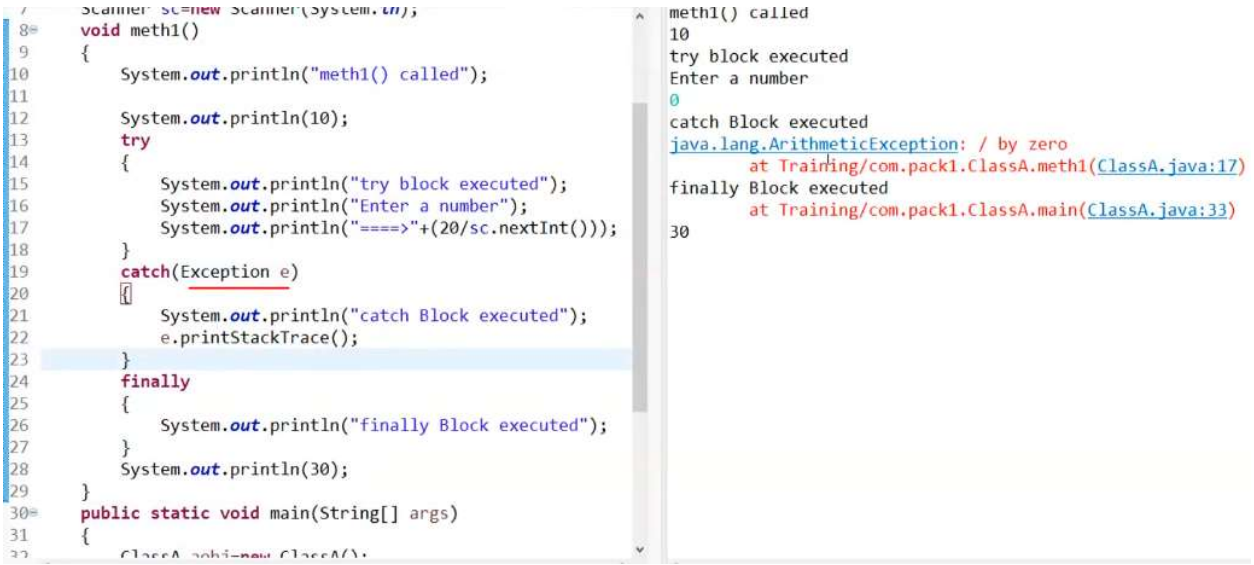
```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(NullPointerException e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA obj=new ClassA();
```

```
meth1() called
10
try block executed
Enter a number
0
finally Block executed
Exception in thread "main" java.lang.ArithmeticException: / by z
        at Training/com.pack1.ClassA.meth1(ClassA.java:17)
        at Training/com.pack1.ClassA.main(ClassA.java:33)
```



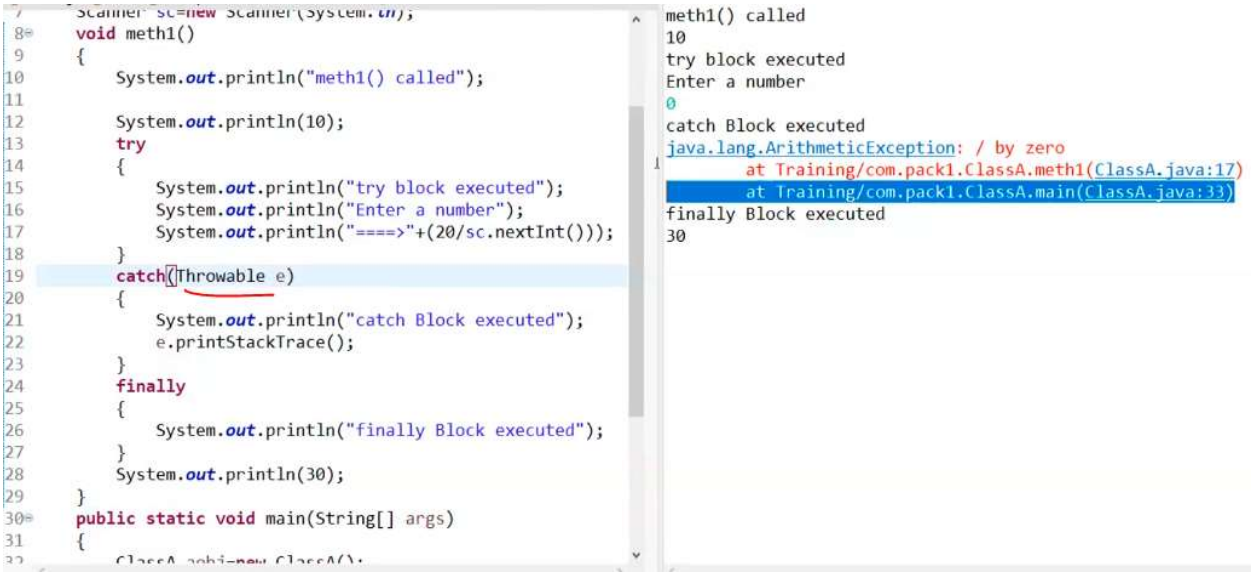
```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(RuntimeException e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA obj=new ClassA();
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
        at Training/com.pack1.ClassA.meth1(ClassA.java:17)
        at Training/com.pack1.ClassA.main(ClassA.java:33)
finally Block executed
30
```



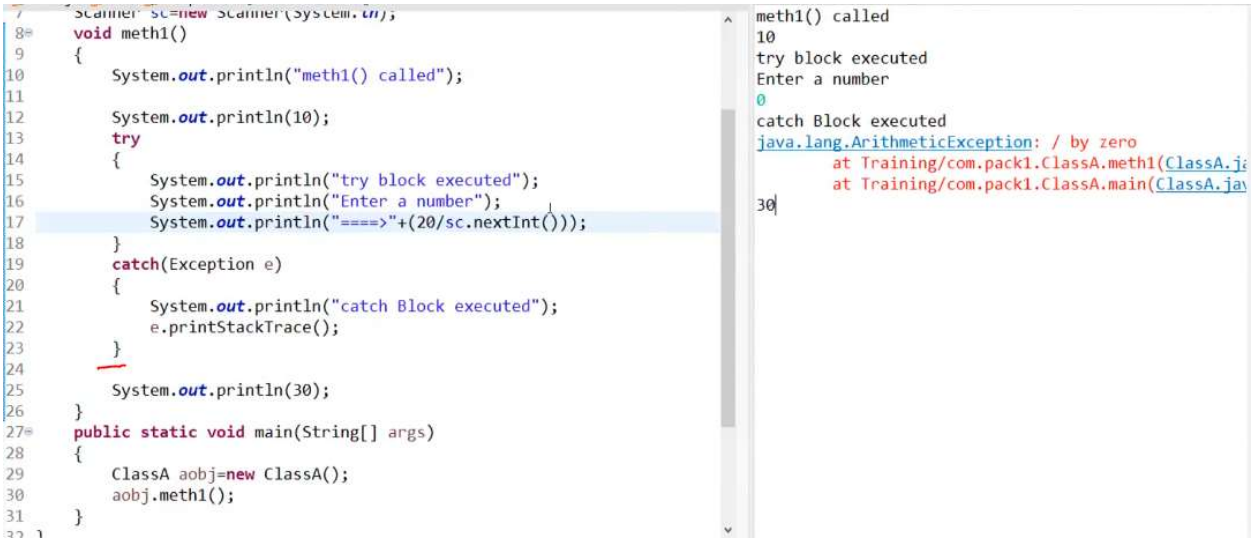
```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(Exception e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA obj=new ClassA();
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
        at Training/com.pack1.ClassA.meth1(ClassA.java:17)
finally Block executed
        at Training/com.pack1.ClassA.main(ClassA.java:33)
30
```



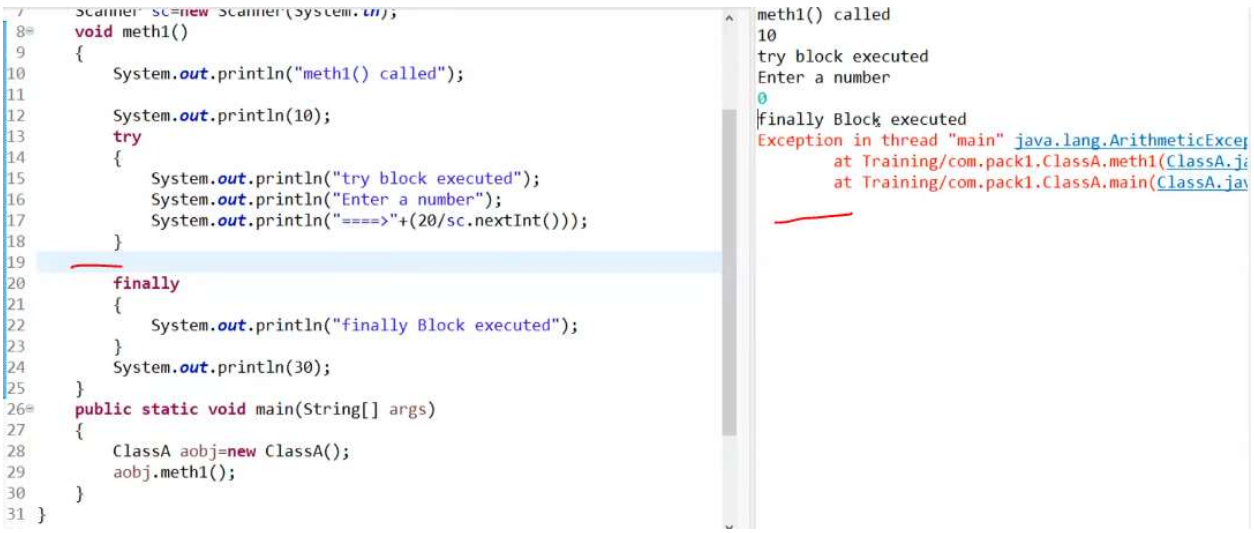
```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(Throwable e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }
    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA aobj=new ClassA();
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
        at Training/com.pack1.ClassA.meth1(ClassA.java:17)
        at Training/com.pack1.ClassA.main(ClassA.java:33)
finally Block executed
30
```



```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }
    catch(Exception e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }

    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA aobj=new ClassA();
    aobj.meth1();
}
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
        at Training/com.pack1.ClassA.meth1(ClassA.ja
        at Training/com.pack1.ClassA.main(ClassA.jav
30
```



```
Scanner sc=new Scanner(System.in);
void meth1()
{
    System.out.println("meth1() called");

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("====>"+(20/sc.nextInt()));
    }

    finally
    {
        System.out.println("finally Block executed");
    }
    System.out.println(30);
}
public static void main(String[] args)
{
    ClassA aobj=new ClassA();
    aobj.meth1();
}
}
```

```
meth1() called
10
try block executed
Enter a number
0
finally Block executed
Exception in thread "main" java.lang.ArithmeticExcep
        at Training/com.pack1.ClassA.meth1(ClassA.ja
        at Training/com.pack1.ClassA.main(ClassA.jav
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
        Scanner sc=new Scanner(System.in);

        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("====>"+(20/sc.nextInt()));
            System.out.println("Hello world!!!");
            sc.close();
        }
        catch(Exception e)
        {
            System.out.println("catch Block executed");
            e.printStackTrace();
        }
        finally
        {
            System.out.println("finally Block executed");
        }
```

```
meth1() called
10
try block executed
Enter a number
5
====>4
Hello world!!!
finally Block executed
30
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
        Scanner sc=new Scanner(System.in);

        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("====>"+(20/sc.nextInt()));
            System.out.println("Hello world!!!");
            sc.close();
        }
        catch(Exception e)
        {
            System.out.println("catch Block executed");
            e.printStackTrace();
        }
        finally
        {
            System.out.println("finally Block executed");
        }
```

```
meth1() called
10
try block executed
Enter a number
0
catch Block executed
java.lang.ArithmeticException: / by zero
finally Block executed
        at Training/com.pack1.ClassA.meth1(ClassA.ja
        at Training/com.pack1.ClassA.main(ClassA.jav
30
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
        Scanner sc=new Scanner(System.in);

        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("====>"+(20/sc.nextInt()));
            System.out.println("Hello world!!!");
        }
        catch(Exception e)
        {
            System.out.println("catch Block executed");
            e.printStackTrace();
            sc.close();
        }
        finally
        {
            System.out.println("finally Block executed");
        }
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("meth1() called");
        Scanner sc=new Scanner(System.in);

        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("====>"+(20/sc.nextInt()));
            System.out.println("Hello world!!!");
        }
        catch(Exception e)
        {
            System.out.println("catch Block executed");
            e.printStackTrace();
        }
        finally
        {
            System.out.println("finally Block executed");
            sc.close();
        }
```

```java
    void meth1()
    {
        System.out.println("meth1() called");
        Scanner sc=new Scanner(System.in);

        String arr[]=new String[5];
        arr[1]="Kishan";
        arr[2]="Java";

        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("1)====>"+(20/sc.nextInt()));
            System.out.println("Enter index postion of array");
            System.out.println("2)===>"+arr[sc.nextInt()].toUpperCase());
            System.out.println("Hello world!!!");
        }
        catch(Exception e)
        {
            System.out.println("catch Block executed");
            e.printStackTrace();
        }
        finally
```

```
meth1() called
10
try block executed
Enter a number
5
1)====>4
Enter index postion of array
1
2)===>KISHAN
Hello world!!!
finally Block executed
30
```

```java
void meth1()
{
    System.out.println("meth1() called");
    Scanner sc=new Scanner(System.in);

    String arr[]=new String[5];
    arr[1]="Kishan";
    arr[2]="Java";

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("1)====>"+(20/sc.nextInt()));
        System.out.println("Enter index postion of array");
        System.out.println("2)===>"+arr[sc.nextInt()].toUpperCase());
        System.out.println("Hello world!!!");
    }
    catch(ArithmeticException e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }
    finally
```

```
meth1() called
10
try block executed
Enter a number
5
1)====>4
Enter index postion of array
0
finally Block executed
Exception in thread "main" java.lang.NullPoi
        at Training/com.pack1.ClassA.meth1(C
        at Training/com.pack1.ClassA.main(Cl
```

```java
void meth1()
{
    System.out.println("meth1() called");
    Scanner sc=new Scanner(System.in);

    String arr[]=new String[5];
    arr[1]="Kishan";
    arr[2]="Java";

    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("1)====>"+(20/sc.nextInt()));
        System.out.println("Enter index postion of array");
        System.out.println("2)===>"+arr[sc.nextInt()].toUpp
        System.out.println("Hello world!!!");
    }
    catch(NullPointerException e)
    {
        System.out.println("catch Block executed");
        e.printStackTrace();
    }
    finally
```

```
meth1() called
10
try block executed
Enter a number
0
finally Block executed
Exception in thread "main" java.lang.ArithmeticException: /
        at Training/com.pack1.ClassA.meth1(ClassA.java:21)
        at Training/com.pack1.ClassA.main(ClassA.java:41)
```

```java
    System.out.println(10);
    try
    {
        System.out.println("try block executed");
        System.out.println("Enter a number");
        System.out.println("1)====>"+(20/sc.nextInt()));
        System.out.println("Enter index postion of array");
        System.out.println("2)===>"+arr[sc.nextInt()].toUpp
        System.out.println("Hello world!!!");
    }
    catch(ArithmeticException e)
    {
        System.out.println("1st catch Block executed");
        e.printStackTrace();
    }
    catch(NullPointerException e)
    {
        System.out.println("2nd catch Block executed");
        e.printStackTrace();
    }
    finally
    {
        System.out.println("finally Block executed");
        sc.close();
    }
```

```
meth1() called
10
try block executed
Enter a number
5
1)====>4
Enter index postion of array
0
2nd catch Block executed
java.lang.NullPointerException: Cannot invoke "String.toUppe
        at Training/com.pack1.ClassA.meth1(ClassA.java:23)
        at Training/com.pack1.ClassA.main(ClassA.java:46)
finally Block executed
30
```

```
        System.out.println(10);
        try
        {
            System.out.println("try block executed");
            System.out.println("Enter a number");
            System.out.println("1)====>"+(20/sc.nextInt()));
            System.out.println("Enter index postion of array");
            System.out.println("2)===>"+arr[sc.nextInt()].toUpp
            System.out.println("Hello world!!!");
        }
        catch(ArithmeticException e)
        {
            System.out.println("1st catch Block executed");
            e.printStackTrace();
        }
        catch(NullPointerException e)
        {
            System.out.println("2nd catch Block executed");
            e.printStackTrace();
        }
        finally
        {
            System.out.println("finally Block executed");
            sc.close();
        }
```

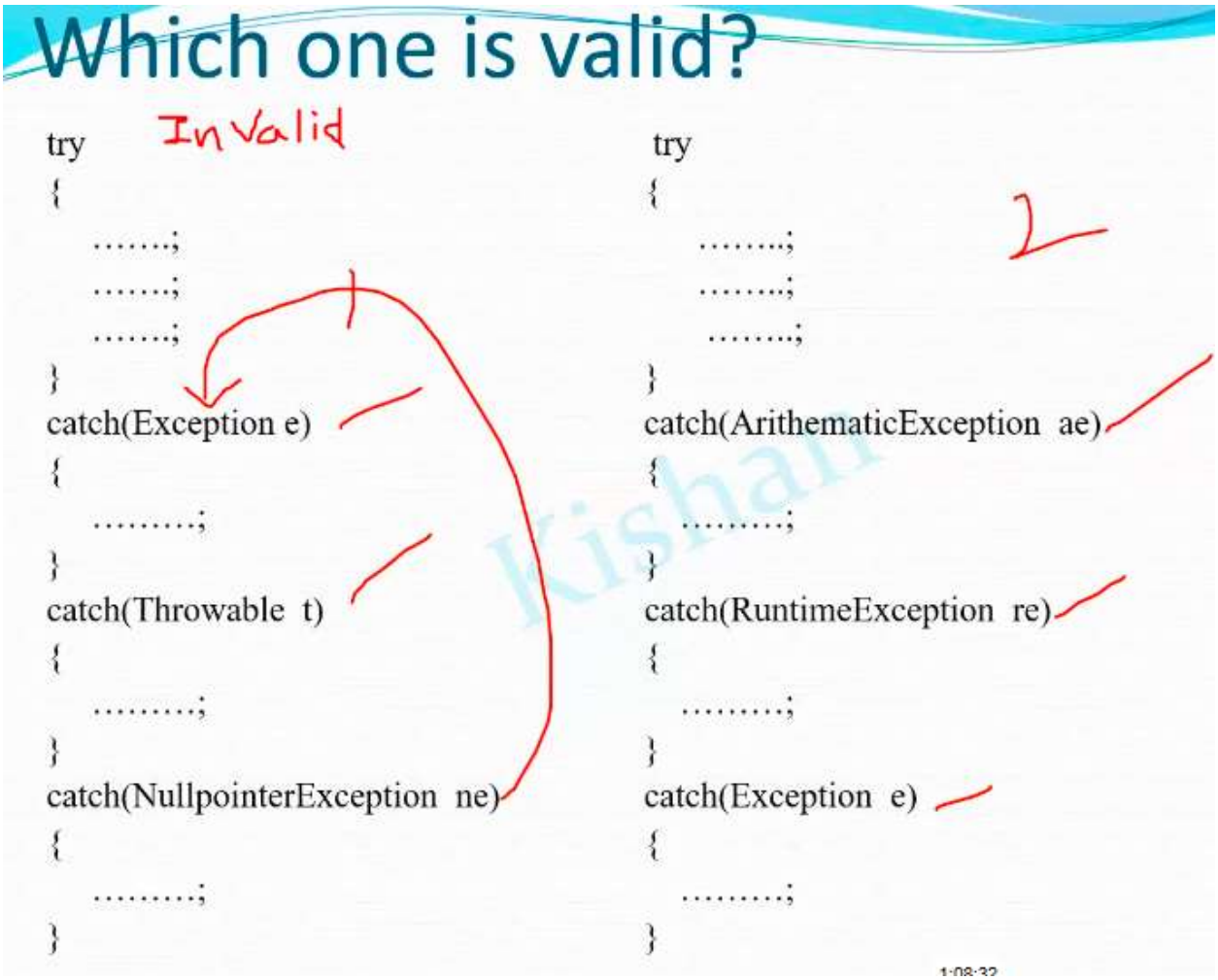
```
meth1() called
10
try block executed
Enter a number
0
1st catch Block executed
java.lang.ArithmeticException: / by zero
        at Training/com.pack1.ClassA.meth1(ClassA.java:21)
        at Training/com.pack1.ClassA.main(ClassA.java:46)
finally Block executed
30
```

Which one is valid?

```
try
{
    .......;
    .......;
    .......;
}
catch(Exception e)
{
    .........;
}
catch(Throwable  t)
{
    .........;
}
catch(NullpointerException  ne)
{
    .........;
}
```

```
try
{
    .........;
    .........;
     .........;
}
catch(ArithematicException  ae)
{
    .........;
}
catch(RuntimeException  re)
{
    .........;
}
catch(Exception  e)
{
    .........;
}
```

Which one is valid?



```
16          System.out.println(10);
17          try
18          {
19              System.out.println("try block executed");
20              System.out.println("Enter a number");
21              System.out.println("1)====>"+(20/sc.nextInt()));
22              System.out.println("Enter index postion of array");
23              System.out.println("2)===>"+arr[sc.nextInt()].toUpperCase());
24              System.out.println("Hello world!!!");
25          }
26          //System.out.println("hi"); // C.E
27          catch(NullPointerException e)
28          {
29              System.out.println("1st catch Block executed");
30              e.printStackTrace();
31          }
32          catch(Exception e)
33          {
34              System.out.println("2nd catch Block executed");
35              e.printStackTrace();
36          }
37          //System.out.println("hi"); // C.E
38          finally
39          {
40              System.out.println("finally Block executed");
```

Keypoint in Exception Handling:

try-finally-catch ⇨ Invalid
finally-try-catch ⇨ Invalid
catch-try-finally ⇨ Invalid
try-catch-finally ⇨ Valid

1) We can handle an exception by using try, catch, finally blocks.
2) Whenever we are using all the **three** blocks we should 100% maintain the order.
3) Inside try block always we need **to write minimum code**. [Write **only** suspecious code inside the try block]
4) If there is an exception occurred in the try block then immediately the compiler will be coming to **its respective catch block**. Remaining code which is present inside the try block will not be executed.
5)A catch block will be executed only if there is an exception occurred in the try block and we are catching that respective exception.
6) If we are catching the parent Exception of all the exception classes ie., **[Exception]** then every Exception will be handled.
7) **A single try block never exists.**
8) try block should be followed with either catch block (or) finally block (or) both.

try-catch ⇨ Valid | try-finally ⇨ Valid | catch-finally ⇨ Invalid | try-catch-finally ⇨ Valid | try ⇨ Invalid

9) If we are not writing catch block in our program then we will not be having any error but if there is an exception occurred in our program it will not be handled.
10) finally block is used to close the existing database/server connections.
11) Between **try-catch-finally** blocks there should not be any individual statements.
12) **We can handle multiple exceptions by using multiple catch blocks.**
13) For a single try block we can write multiple catch blocks but we need to write a single finally block
14) Multiple catch blocks are allowed but multiple finally blocks are not allowed.
15) **Whenever we are using multiple catch blocks always parent exception should be handled in the last catch block.**
16) If we are using multiple catch blocks duplicate exception handling is not allowed. [We will be getting an Compile time error]
17) From Java 1.7v onwards we can write a single try block also. try(Resources){ }
18) From Java 1.7v onwards we can handle multiple exceptions by using a single cacth block. But those exceptions should not have any parent child relation ship.