

Finally keyword

finally : finally is keyword in exception handling which always execute if exception generate program or not. **Normally finally block is used for resource cleaning purpose like as for database connection close,file close etc**

```
try
{
    write here exception logic
}
finally
{
    write here logic those want to execute always
}
```

Or we can write try catch and finally at time.

```
try
{
    write here exception logic
}
catch(Exceptiontype ref)
{
    write here your logics
}
finally
{
    write here logic those want to execute always
}
```

Q. can you write try without catch?

Yes we can write the try without catch by using finally.

```
import java.util.*;
public class TestFinallyApp
{
    public static void main(String x[])
    {
        int a,b,c;
        Scanner xyz = new Scanner(System.in);
        try
        {
            System.out.println("Enter the two values");
            a=xyz.nextInt();
            b=xyz.nextInt();
            c=a/b;
            System.out.printf("Division is %d\n",c);
        }
        finally
        { System.out.println("I can execute always");
        }
        System.out.println("Logic1");
        System.out.println("Logic2");
        System.out.println("Logic3");
    }
}
```

Output

```

Enter the two values
8
2
Division is 4
I can execute always
Logic1
Logic2
Logic3

C:\Program Files\Java\jdk1.8.0_291\bin>java TestFinallyApp
Enter the two values
9
0
I can execute always
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at TestFinallyApp.main(TestFinallyApp.java:13)

C:\Program Files\Java\jdk1.8.0_291\bin>

```

Note: finally cannot handle the exception finally only execute its own block in any situation. If we want to handle the exception in java we have to write catch block.

Q. what is the diff between catch and finally ?

Catch is responsible to handle the exception and manage the other program but finally cannot handle the exception just execute its own block and if exception generate in program then finally get executed before exception .

Source Code

```

import java.util.*;
public class TestFinallyApp
{
    public static void main(String x[])
    {
        int a,b,c;
        Scanner xyz = new Scanner(System.in);
        try

```

```
{
    System.out.println("Enter the two values");
    a=xyz.nextInt();
    b=xyz.nextInt();
    c=a/b;
    System.out.printf("Division is %d\n",c);
}
catch(Exception ex)
{ System.out.println("Error is "+ex);
}
finally
{ System.out.println("I can execute always");
}
System.out.println("Logic1");
System.out.println("Logic2");
System.out.println("Logic3");
}
}
```

Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac TestFinallyApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java TestFinallyApp
Enter the two values
9
0
Error is java.lang.ArithmeticException: / by zero
I can execute always
Logic1
Logic2
Logic3

C:\Program Files\Java\jdk1.8.0_291\bin>_
```

Q. What is the diff between final, finally and finalize ?

Final is keyword we can use with variable, function and class

Final keyword work with variable for constant declaration purpose, for method avoids the method overriding and for class for avoids the inheritance.

Finally is block use with try block in exception handling situation for execute the code any time and finalize is method of object class which is used perform garbage collection.

Throws keyword

Throws is keyword in exception handling we can use with function.

Important points Related with throws keyword

- 1) Throws keyword is used for handle the checked exception
- 2) When we use throws keyword then we not need to write try and catch block in function definition we need to write try and catch block at the time of function.

3) When exception generate in function definition it will throw on function calling.

Syntax of throws keyword

```
returntype functionname(arguments)throws exceptionclassname  
{  
    write here logic in function |  
}
```

Note: Normally throws use with function to generate the compile warning related with exception.

Suppose consider we write function and it contain some logic and logic may be generate the exception run time and our function use by some another person then we need to provide the warning to related exception to program at compile time to avoid the run exception in this case we can use the throws with function and handle the checked exception

Means when someone calls our function then compiler generate the compile time warning related with exception.

```

class Div
{
    8    0
    void setValue(int x,int y) throws Exception
    {
        int z = x/y; // 8/0
        System.out.println("Division is "+z);
    }
}
public class DivApplication
{
    public static void main(String x[])
    {
        try
        {
            Div d = new Div();
            d.setValue(8,0);
        }
        catch(Exception ex)
        {
            System.out.println("Error is "+ex);
        }
    }
}

```

Output

```

C:\Program Files\Java\jdk1.8.0_291\bin>javac DivApplication.java

C:\Program Files\Java\jdk1.8.0_291\bin>java DivApplication
Error is java.lang.ArithmeticException: / by zero

C:\Program Files\Java\jdk1.8.0_291\bin>_

```

throw keyword

throw is keyword or clause which is specially design for handle the user defined exception.

Q. what is the user defined exception?

User define exception means those exception defined by user for its own use called as user defined exception.

Q. Why we need to use user defined exception?

1) If programmer want to create the customize error message and exception according to his application then he can write the user defined exception.

2) If programmer having logical error at run time and java API not provide the appropriate classes to handle that logical error then user can create own exception class and handle the error then he need to create user defined exception

Q. How To Create User Defined Exceptions?

If we want to create the user defined exceptions in java we have the some important steps.

Step1

Create the class and inherit the any exception class in it.

```
class VoterException extends ArithmeticException
{
}
```

Step2:

Define the function under the class and write your own logic

```
class VoterException extends ArithmeticException
{
    int age;
    String getVoterAge(int age)
    {
        this.age=age;
    }
}
```



```
String errorMessage()  
{  
    if(age<18)  
        { return "you are not eligible for voting";  
        }  
    if(age<0)  
        { return "Invalid Age ";  
        }  
}
```

Step3

If we want to handle the user defined exception then we need to create the object of user defined exception class and throw it and pass the reference of user exception class in catch block where we call it.

