**The Class Throwable: -** Class Throwable is super class of all the exception classes. It contains some public methods that are inherited by all exception classes in java.

**1.String getMessage():-**Define the reason of any exception.

**2.void printStackTrace():-**This method prints exception in the following format:-

Name Of Exception: - description followed by stack trace.

**3.String toString():-**It prints exception information in the following format:-

Name of Exception: - description

**Example:**-

try

{

System.out.println(10/0);

}

Catch(ArithmeticException ae)

{

System.out.println(ae.getMessage());

System.out.println(ae.toString());

System.out.println(ae.printStackTrace());

}

**Note:-**default exception handler internally uses printStackTrace() method.

* **Throw and throws keyword:-**
  + **Throw:-**
    - * + Java throw keyword used to explicitly throw an exception.
        + Checked exception cannot be propagated using throw keyword.
        + Throw is used within the method.
        + Throw is followed by an instance.
* **Throws:-**
  + - * + Java throws keyword used to declare exception.
        + Checked exception can be propagated with throws.
        + Throws is followed by class.
        + Throws is used with the method signature.
        + You can throw multiple exception.

**Example:-**

Public classs MyException

{

Public static int sum(int num1,int num2)

{

If(num1==0)

{

Throw new ArithmeticException(“Number is not valid”);

}

Else

{

Return num1+num2;

}

}

Public static void main(Stirng []args)

{

Try

{

Int res=sum(0,10);

System.out.println(“Result = ”+res);

}

Catch(ArithmeticException ae)

{

System.out.println(ae.getMessage());

}

}

}

**Exceptoin Handling Keyowrds Summary:-**

1. **Try:-**To maintain risky code.
2. **Catch:**- To maintain handling code.
3. **Finally:**-To maintain clean-up code.
4. **Throw:**-