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

Important points to remember

- Default exception handler can handle only one exception at a time and that is the most recently raised exception
- There should not be any statements b/w try, catch and finally.
- We can handle multiple exceptions by writing multiple catch blocks.
- A single try block can be followed by several catch blocks.
- Catch block does not always exit without a try, but a try block exit without a catch block.
- Finally block is always executed whether there is an exception or not.
- There should be only one finally block for a try block.
- We should follow try-catch-finally order.
- Until 1.6 version try should be followed by either catch or finally but 1.7 version we can take only try with resource without catch or finally

  try(FileInputStream input = new FileInputStream("file.txt"))
- From 1.7 we can use multiple catch blocks in one statement only

Which one is valid?

```
try                                       try
{                                         {
   …….;                                      …….;
   …….;                                      …….;
   …….;                                       …….;
}                                         }
catch(Exception e)                        catch(ArithematicException  ae)
{                                         {
   ………;                                      ………;
}                                         }
catch(Throwable  t)                       catch(RuntimeException  re)
{                                         {
   ………;                                      ………;
}                                         }
catch(NullpointerException  ne)           catch(Exception  e)
{                                         {
   ………;                                      ………;
}                                         }
```