Exception Handling -

- Exception is an error event that can happen during the execution of a program and disrupt its normal flow.
- Exceptions in Java can arise from different kind of situations such as wrong data entered by user, H/w failure, network connection failure, Database server down etc.
- Java provide a robust and object oriented way to handle exception scenarios, known as Java Exception Handling.
- Whenever an error occurs, it creates an exception object. The exception object contain a lot of debugging infn such as method hierarchy, line number, and type of exception etc.
- The normal flow of the program halts and JRE tries to find someone than can handle the raised exception.
- When the exception occurs in a method, the process of creating the exception object & handling it over to runtime environment is called throwing the exception

Catching the Exception -

- Once runtime receives the exception object, it tries to find the handler for the exception. Exception handler is the block of code that can process the exception object.

- If methods call stack is A→B→C & exception is raised in C, then the search for appropriate handler will move from C→B→A.

- If appropriate exception handler is found, exception object is passed to it. The handler is said to be catching the exception.

* finally block -
  - Optional block.
  - If written, will execute at any cost.
  - Written at the end of try-catch block.

Object

↓

Throwable

- Error
  - → Thread Death
  - → Virtual Machine Error
    - ← Out of Memory Error
    - ← StackOverflow Error
  - → IOError
- Exception
  - → RuntimeException
  - → IO Exception
  - → SQLException
  - → Broken Barrier Exception

RuntimeException
- → NullPointer
- → Arithmetic.

** Black Pen → Unchecked / Runtime Exception
Blue Pen → Checked / Compile Time Exception

Throw Keyword-

- Exception object is required with it.
  for example-

```
boolean b=true;
if(b) {
    throw new ArithmeticException();
}
```

- The throw keyword is used to explicitly throw an exception from a method or any block of code.

Throws Keyword-
It is used in the signature of method to indicate that this method might throw one of the exception. The caller to these method has to handle the exception using a try-catch block.

For example-

```
try {
        fun2();
} catch (Exception e) {
      System.out.println(e.getMessage());
  }

static void fun2() throws ArithmeticException {
        boolean b=true;
        if (b) {
            throw new Exception ("Fault");
        }
}
```

Try-Catch block-

```
try {
        Exception/Error block.
} catch (<Type of Exception Object>) {
      ____
      ____
  }
Optional block { finally {
```