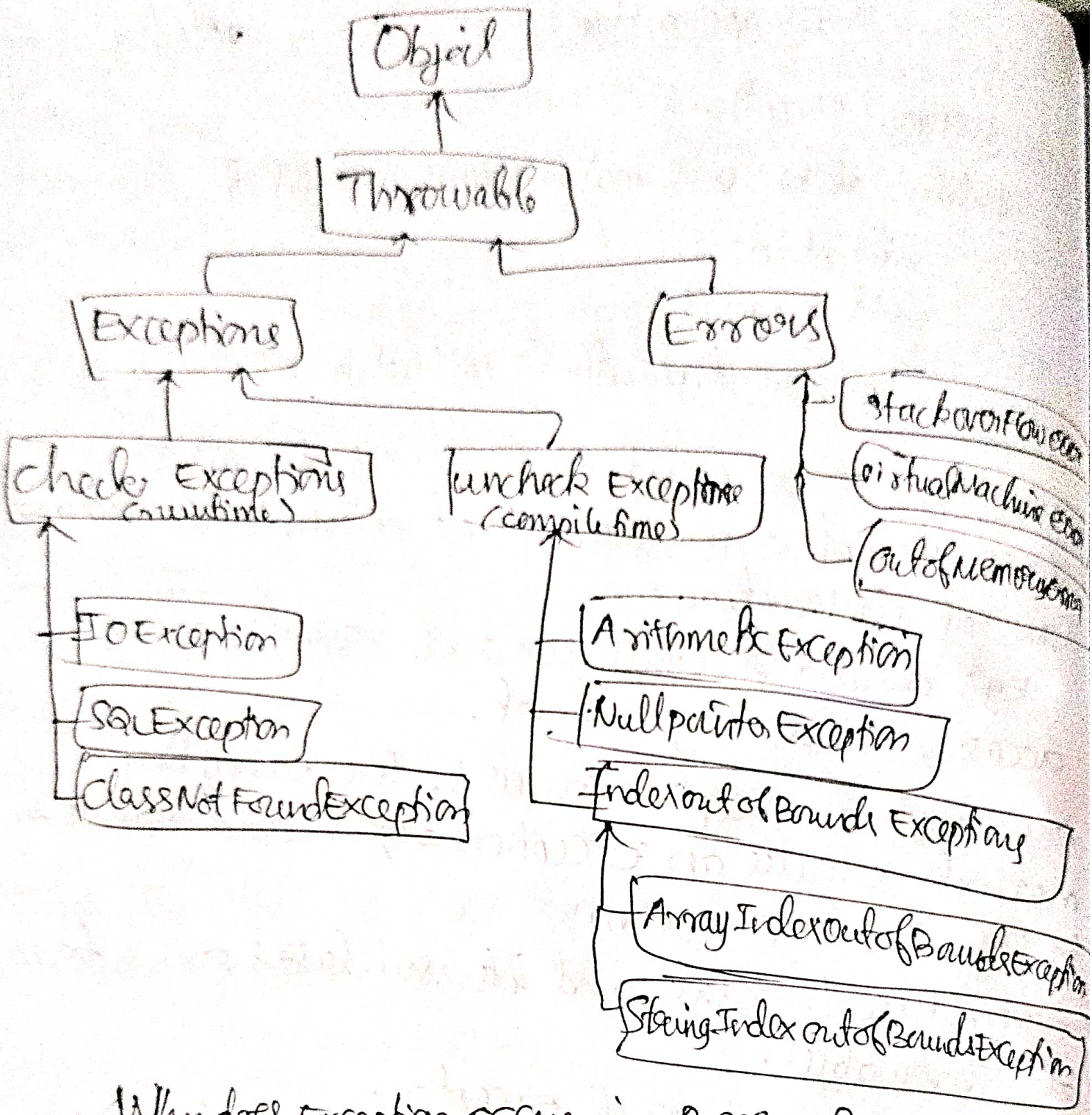
Exception Handling: 25/6/24

```
class Without Exception {
    public static void main (String args[]) {
        int d=0;
        int a = 42/d;
        System.out.println ("Will not be printed:");
```

* Exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions:
  Eg: Hard disk crash, out of bounds array access, Divide by zero etc.
* When an exception occurs, the executing method creates an Exception object and hands in to the runtime system
* The Exception class and its subclasses are a form of Throwables.
* There are 2 types of exception:
  (i) Checked Exceptions: Checked exceptions are called compile-time exceptions because these exceptions are checked at compile-time by the compiler
  (ii) unchecked Exceptions: The unchecked exceptions are just opposite to the checked exceptions. The compiler will not check these exceptions at compile time.



Why does Exception occur in Program?

* Opening a non-existing file in your program
* Reading a file from a disk, but the file doesn't exist there
* Writing data to a disk but the disk is full or unformatted
* When the program asks for user input and the user enters invalid data.
* When a user attempts to divide an integer value/0, an exception occurs.

* When a data streams is in an invalid format.

Errors.

* Errors are usually beyond the control of the programmer and we should not try to catch
* Instances of error are internal errors in Java runtime environment. These are ~~rare~~ rare and usually fatal and ∴ not supposed to be handled by a program.
* An error indicates a serious problem that a reasonable application should not try to catch
* Instances of error are thrown, when the Java virtual machine faces some memory leakage problem, insufficient memory problem, dynamic linking failure or whensome other "hard" failure in the virtual machine occurs.

Keywords for handling Exceptions.

try: This marks the start of a block associated with a set of exception handlers.
catch: The control moves here if an exception is generated.
finally: This is called irrespective of whether an exception has occured or not.
throws: This describes the exceptions which can be raised by a method.

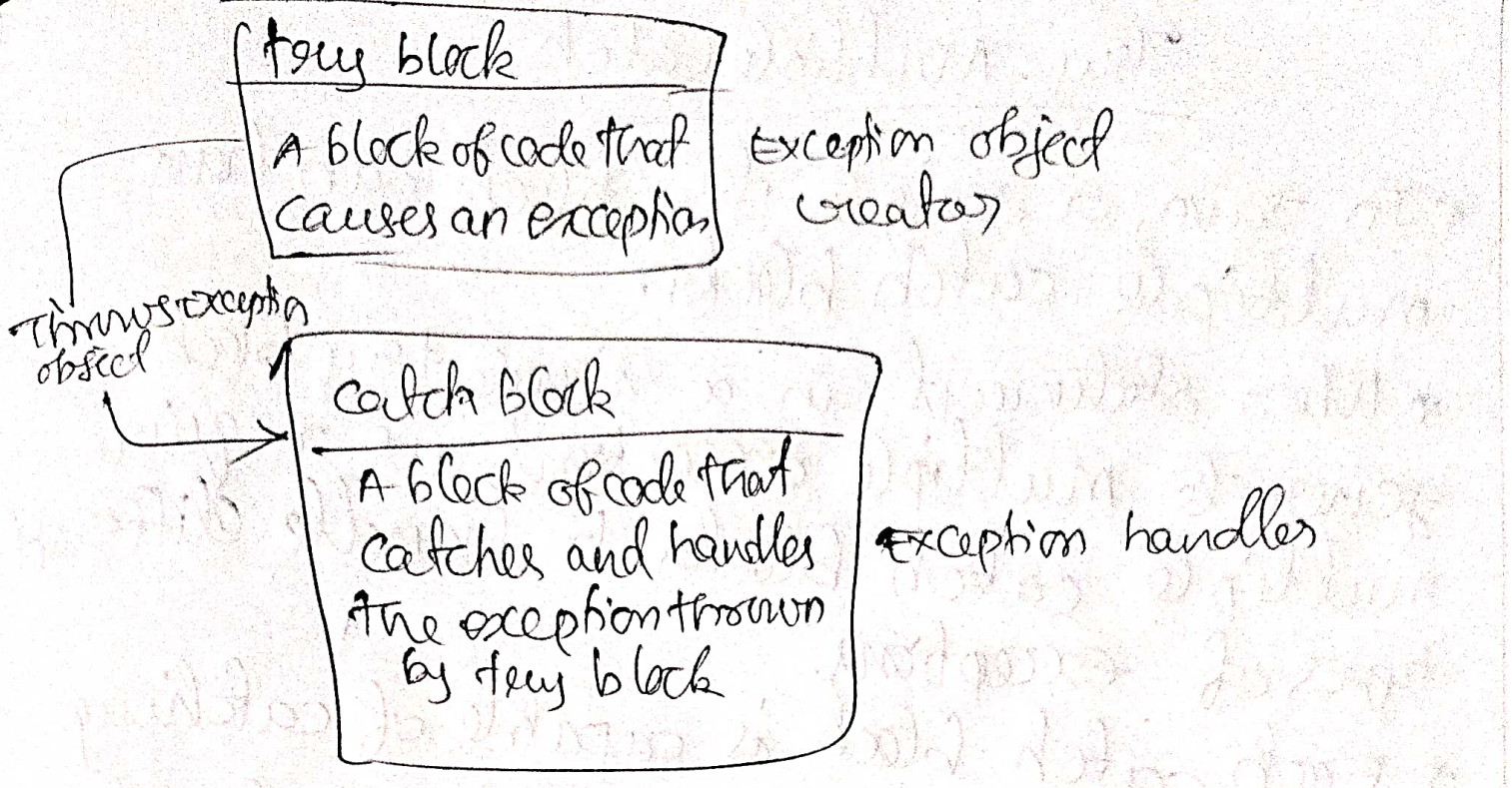
throw: This raises an exception to the first available handler in the call stack, unwinding the stack along the way.

Try-catch:

* A keyword "try" is a block of code of statements that might throw an exception. That's why a try block is also known as exception generated block.
* The Java code that may generate an exception during the execution of program must be placed within a try block.
* A keyword "catch" is a block of code that handles the exception thrown by the try block. That's why it is also known as exception handler block.
* A catch block that catches an exception, must be followed by try block that generates an exception.

Syntax:

```
try
{ // A block of code; // generates an exception
}
catch (exception_class var)
{ // code to be executed when an exception is
    thrown
}
```




```
try
{
    statement 1;
    statement 2;
    statement 3;
}
catch(exception_class var)
{
    statement 4;
}
statement 5;
```

exception occured inside try block

control of execution is passed to catch block

After complete execution of catch block, control never goes back to the execution of remaining code within try block

control goes for execution of remaining code in the program.

Try - Multiple Catch

* In Java, a single try block can have multiple catch blocks.
* When statements in a single try block generate multiple exceptions, we require multiple catch blocks to handle different types of exceptions.
* Each catch block is capable of catching a different exception. That is, each catch block must contain a different exception handler.