Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

SAIRAM DIGITAL RESOURCES

YEAR II SEM III

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to CSE, EEE, EIE, ICE, IT)

UNIT NO 3

EXCEPTION HANDLING AND I/O

3.1 Exception -Exception Hierarchy

COMPUTER SCIENCE & ENGINEERING

Exceptions - exception hierarchy

Exceptions Introduction:

- An exception is an unwanted or unexpected event, which occurs during the execution of a program or at run time that disrupts the normal flow of the program's instructions..
- The normal flow of the program is disrupted and the program/Application terminates abnormally is called as exception
- An exception (or exceptional event) is a problem that arises during the execution of a program.
- Exception is an abnormal condition.

Exceptions - exception hierarchy

Exception and Errors:

- Error: An Error indicates serious problem that a reasonable application should not try to catch.
- Error defines problems that are not expected to be caught under normal circumstances by our program. For example memory error, hardware error, JVM error etc.
- Exception: Exception indicates conditions that a reasonable application might try to catch.
- Exceptions are conditions within the code. A developer can handle such conditions and take necessary corrective actions.

Exceptions - exception hierarchy

Exception and Errors:

Error	Exception
java.lang.Error and java.lang.Exception classes are subclasses of java.lang.Throwable class	
Recovering from Error is not possible. The only solution to errors is to terminate the execution.	We can recover from Exception by using either try-catch blocks or throwing exception back to caller.
It is not possible to handle the Errors using try-catch blocks.	Exceptions can be handled using try-catch blocks and can make program flow normal if they happen.
Compiler will not have any knowledge about unchecked exceptions which include **Errors**	compiler will have knowledge about checked Exceptions. Compiler will force you to keep try-catch blocks
Errors belongs to only one category i.e unchecked.	**Exceptions** in java are divided into two categories – checked and unchecked.

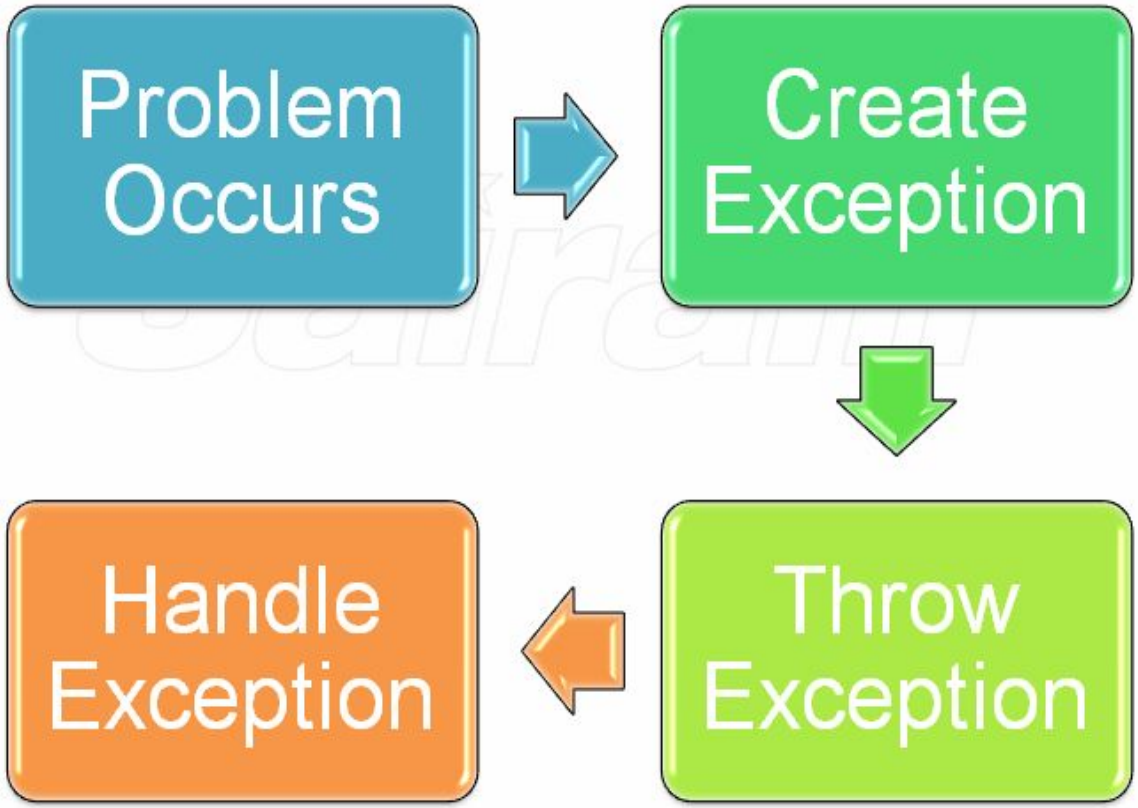
Exceptions - exception hierarchy

Exception Handling:

- The Exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained.
- Therefore these exceptions are needed to be handled.
- Exception Handling is a mechanism to handle runtime errors such as ClassNotFoundException, IOException, SQLException, RemoteException, etc..

Exceptions - exception hierarchy

Exception Handling in Java :



Exceptions - exception hierarchy

Exception Handling:

Exceptions are conditions within the code. A developer can handle such conditions and take necessary corrective actions.

Few examples are

- DivideByZeroException
- NullPointerException
- ArithmeticException
- ArrayIndexOutOfBoundsException

Exceptions - exception hierarchy

Types of Exceptions in Java

These exceptions are caused by user error, programmer error, and physical resources.

Based on these, the exceptions can be classified into three categories.

1. Checked Exception
2. Unchecked Exception
3. Error

Checkedexceptions−An exception that occurs at the compile time, also called as compile time (static time) exceptions. These exceptions cannot be ignored at the time of compilation. So, the programmer should handle these exceptions.

Exceptions - exception hierarchy

Types of Exceptions in Java

- **Unchecked exceptions** − An exception that occurs at run time, also called as Runtime Exceptions. These include programming bugs, such as logic errors or improper use of an API. Runtime exceptions are ignored at the time of compilation.
- **Errors**−Errors are not exceptions, but problems may arise beyond the control of the user or the programmer. Errors are typically ignored in your code because you can rarely do anything about an error. For example, if a stack overflow occurs, an error will arise. They are also ignored at the time of compilation.

Exceptions - exception hierarchy

Types of Exceptions in Java

Checked Exception	Unchecked Exception	Error
The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions.	The classes which inherit RuntimeException are known as unchecked exceptions	Problems may arise beyond the control of the user or the programmer.
Checked exceptions are checked at compile-time.	Unchecked exceptions are not checked at compile-time, but they are checked at runtime.	Error is irrecoverable
Ex:. IOException, SQLException	Ex: ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc.	Ex: OutOfMemoryError, VirtualMachineError, AssertionError etc.

Exceptions - exception hierarchy

Keywords in Java Exception

Keyword	Description
Try	The "try" keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It doesn't throw an exception. It specifies that there may occur an exception in the method. It is always used with method signature.

Exceptions - exception hierarchy

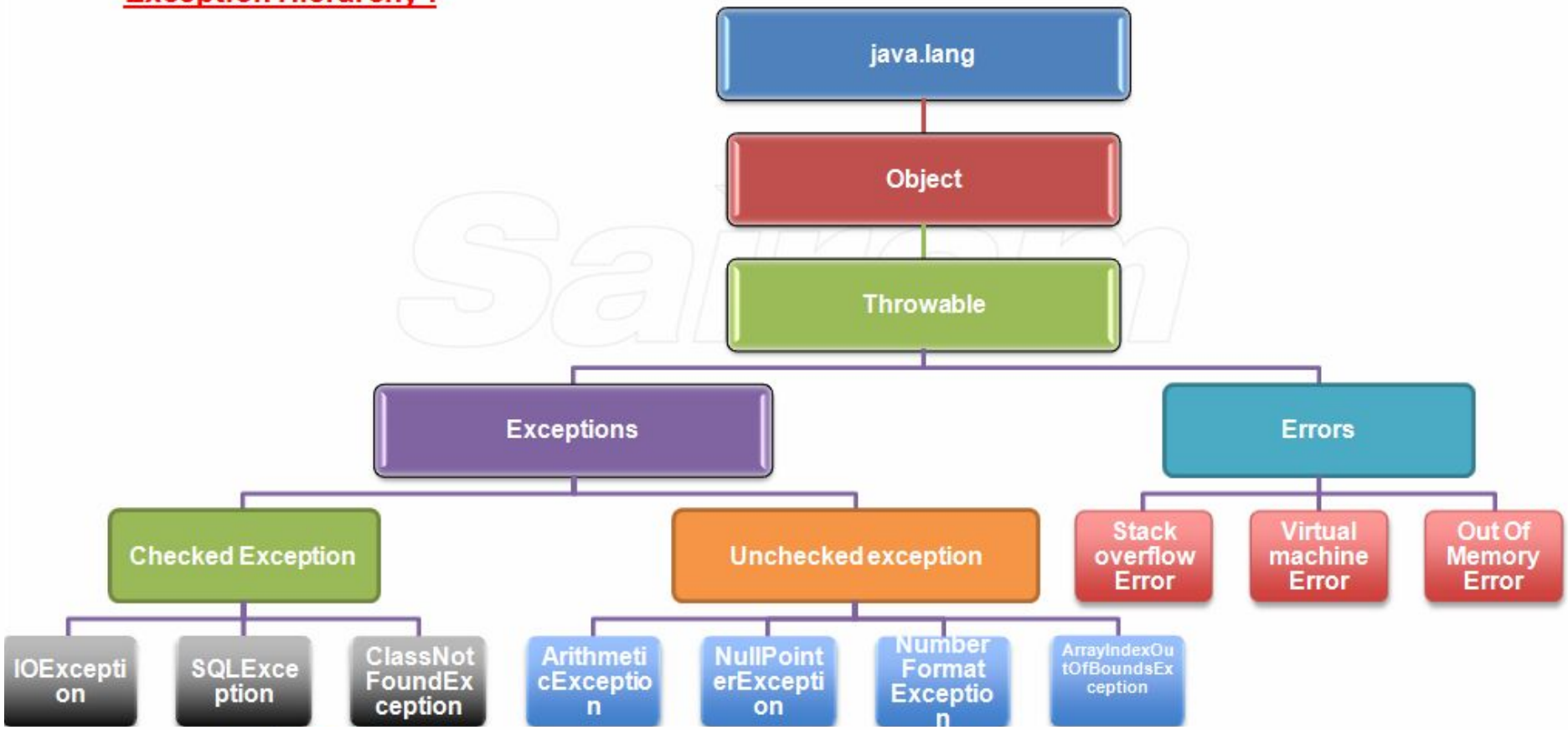
Exception Hierarchy :

 The java.lang.Throwable class is the root class of Java Exception hierarchy which is inherited by two subclasses: Exception and Error.

 All exception classes are subtypes of the java.lang.Exception class. The exception class is a subclass of the Throwable class. Another subclass called Error which is derived from the Throwable class.

Exceptions - exception hierarchy

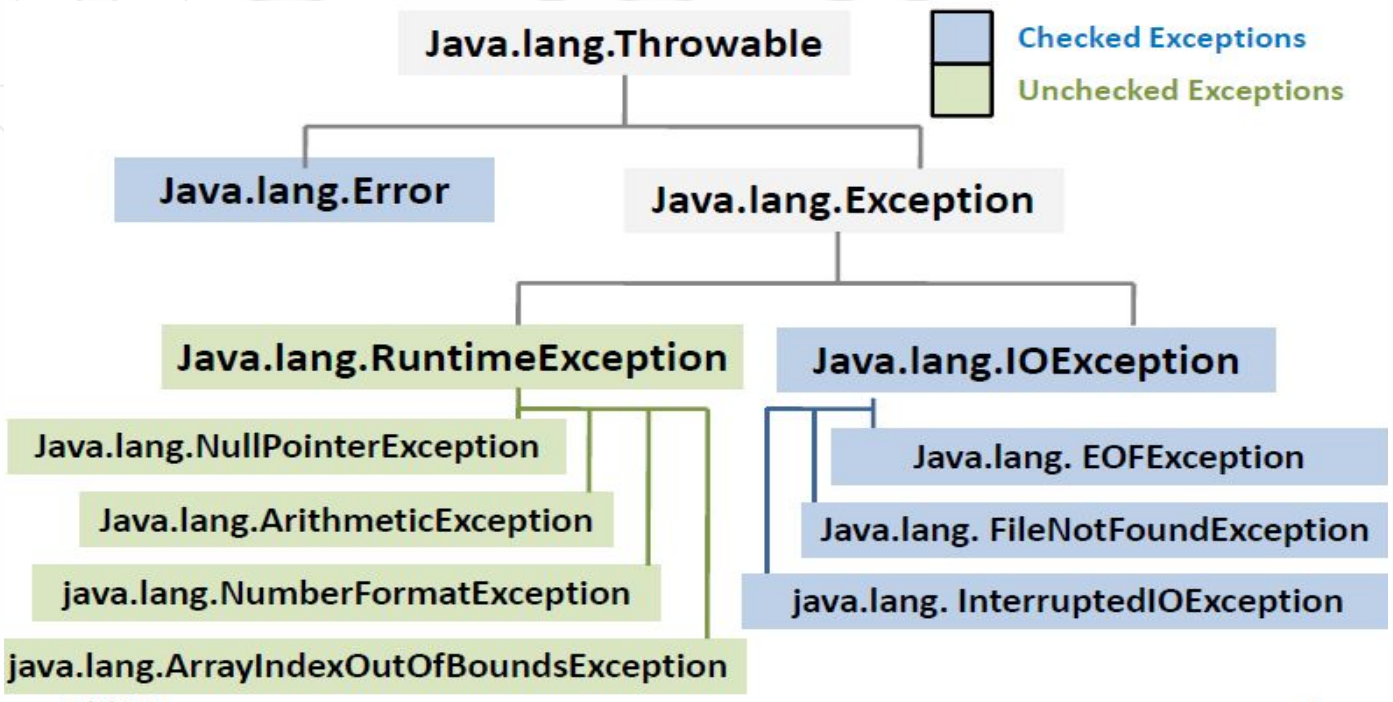
Exception Hierarchy :



Exceptions - exception hierarchy

Exception Hierarchy :

- Normally, programs cannot recover from errors.
- The Exception class has two main subclasses: IOException class and RuntimeException Class.



Exceptions - exception hierarchy

Exception Hierarchy :

- The java.lang.Exception class is the base class for all exception classes.
- All exception and errors types are subclasses of class Throwable, which is base class of hierarchy.
- One branch is headed by Exception.
- This class is used for exceptional conditions that user programs should catch.
- NullPointerException is an example of such an exception.

Exceptions - exception hierarchy

Exception Hierarchy :

- Another branch is Error used by the Java run-time system(JVM) to indicate errors having to do with the run- time environment itself(JRE).
- StackOverflowError is an example of such an error.
- Errors are abnormal conditions that happen in case of severe failures, these are not handled by the Java programs.
- Errors are generated to indicate errors generated by the runtime environment.
- Example: JVM is out of memory.

Exceptions - exception hierarchy

Example of an Exception :

```
public class ex_example {
 public static void main(String[] args){
    int x = 5;
    int y = 0;
    int z = x / y;
  }
}
```

Output

```
java.lang.ArithmeticException: / by zero
    at ExceptionExample.main(ExceptionExample.java:7)
```

Video Link

https://youtu.be/ohpCMpderow

https://youtu.be/5JDZ8CUXLZo