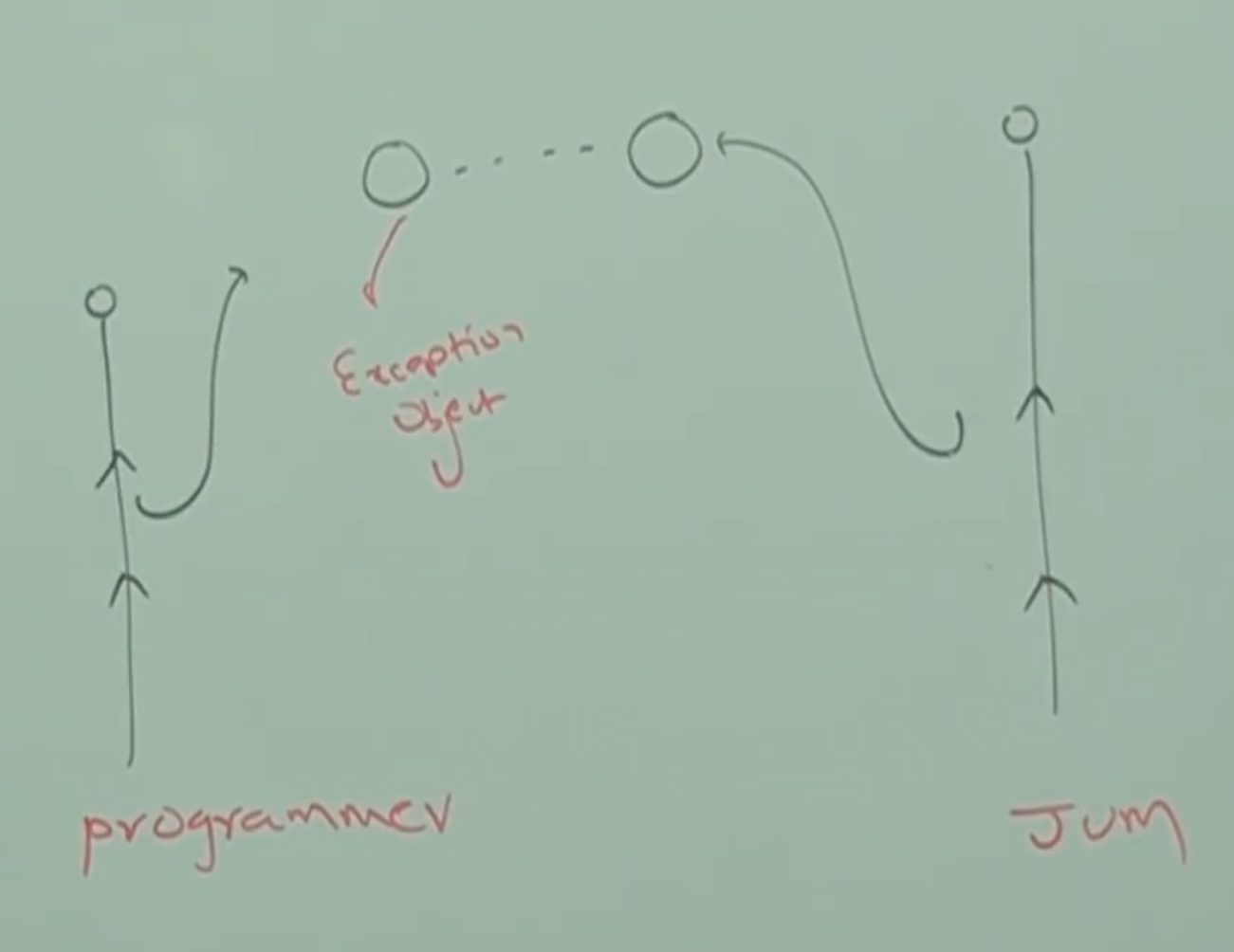
**Exception Handling – Part-06 – throw and throws**

* **Throw Keyword:**



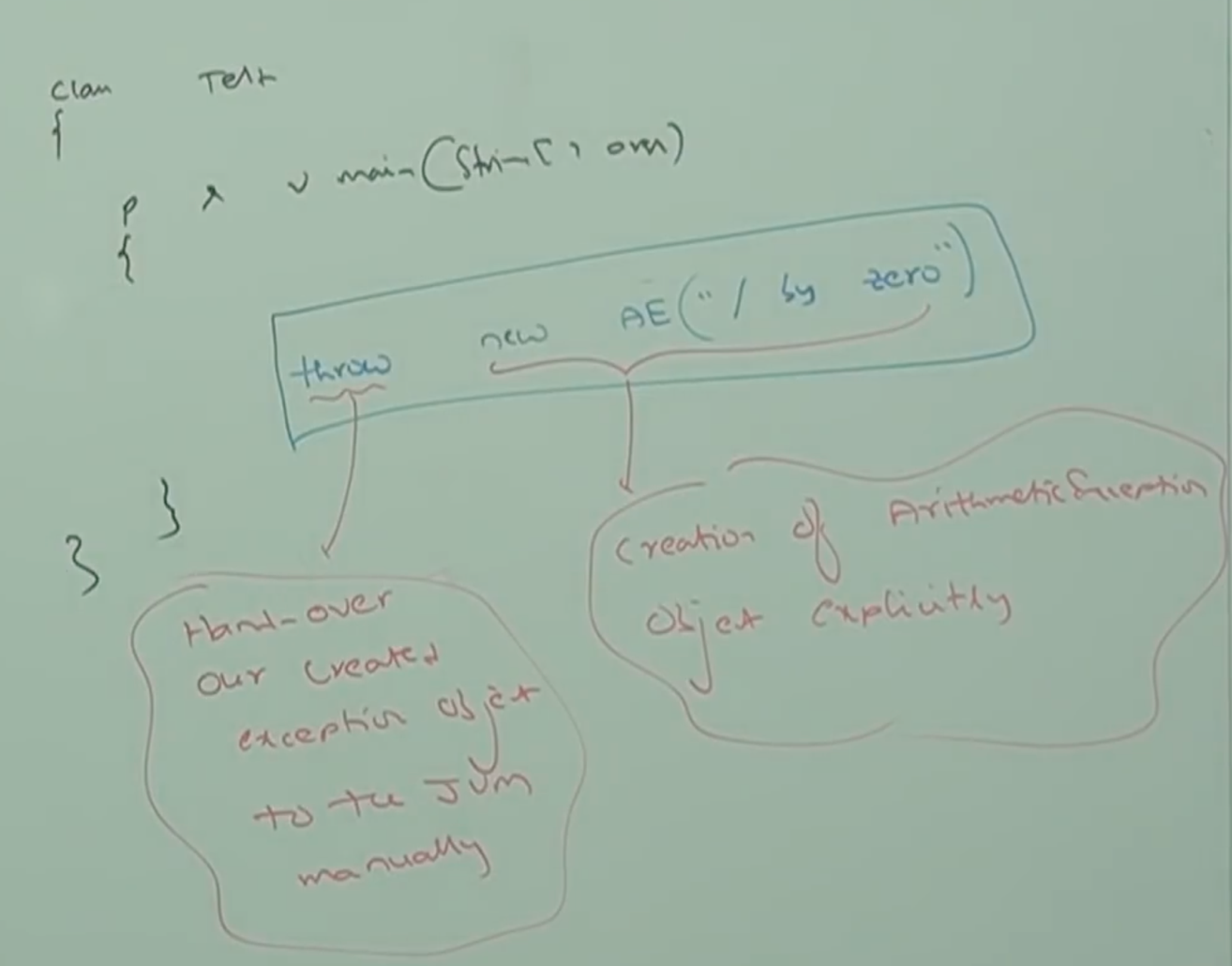
Sometimes we can create exception objects explicitly and we can handover to the JVM manually for this we have to use ***throw*** keyword.

throw new ArithmeticException(“/ by zero”);

(2) (1)

1 - Creation of ArithmeticException explicitly

2 – Handover our created exception object to the JVM manually.



Hence the objective of throw keyword is to handover our created exception object to the JVM manually.

Hence the result of following two programs is exactly same.

class Test{

public static void main(String[] args){

System.out.println(10/0);

}

}

Exception in thread “main” java.lang.ArithmeticException: / by zero

Note: In this case main method is responsible to create exception object and handover to the JVM.

class Test{

public static void main(String[] args){

throw new ArithmeticException(“/ by zero”);

}

}

Exception in thread “main” java.lang.ArithmeticException: / by zero

Note: In this case programmer creating exception object explicitly and handover to the JVM manually.

Note:

Best use of throw keyword is for user defined exceptions or customized exceptions.

* **Case\_01:**

throw e;

If e refers null, then we will get NullPointerException.

Example:

class Test{

static ArithmeticException e = new ArithmeticException();

public static void main(String[] args){

throw e;

}

}

RE: AE

class Test{

static ArithmeticException e;

public static void main(String[] args){

throw e;

}

}

RE: NullPointerException

* **Case\_02:**

After throw statement we are not allowed to write any statement directly, otherwise we will get compile time error saying “unreachable statement”.

class Test{

public static void main(String[] args){

System.out.println(10/0);

System.out.println(“Hello”);

}

}

RE: java.lang.ArithmeticException: / by zero.

class Test{

public static void main(String[] args){

throw new ArithmeticException(“ / by zero”);

System.out.println(“Hello”);

}

}

CE: unreachable statement.

* **Case\_03:**

We can use throw keyword only for throwable types. If we are trying to use for normal Java objects we will get compile time error saying

Incompitable types

class Test{

public static void main(String[] args){

throw new Test();

}

}

CE: incompatible types

found: Test

required: java.lang.Throwable

class Test extends RuntimeException {

public static void main(String[] args){

throw new Test();

}

}

RE: Exception in thread “main” Test

at Test.main();