

# 12 Basic Exception Handling



# Objectives

At the end of the lesson, the student should be able to:

- Define exceptions
- Handle exceptions using a simple **try-catch-finally** block



# Exceptions

- An exception
  - is an event that interrupts the normal processing flow of a program. This event is usually some error of some sort.
  - This causes our program to terminate abnormally.



# Examples

- Some examples of exceptions:
  - `ArrayIndexOutOfBoundsException` exceptions, which occurs if we try to access a non-existent array element
  - `NumberFormatException`, which occurs when we try to pass as a parameter a non-number in the `Integer.parseInt` method.



# Handling Exceptions

- To handle exceptions in Java, we use a **try-catch-finally** block
  - What we do in our programs is that we place the statements that can possibly generate an exception inside this block.



# try-catch-finally block

- The general form of a try-catch-finally block is,

```
try{
    //write the statements that can generate an exception
    //in this block

}
catch( <exceptionType1> <varName1> ){

    //write the action your program will do if an exception
    //of a certain type occurs

}
.
.
.
catch( <exceptionTypen> <varNamen> ){
    //write the action your program will do if an exception
    //of a certain type occurs

}
finally{
    //add more cleanup code here
}
```

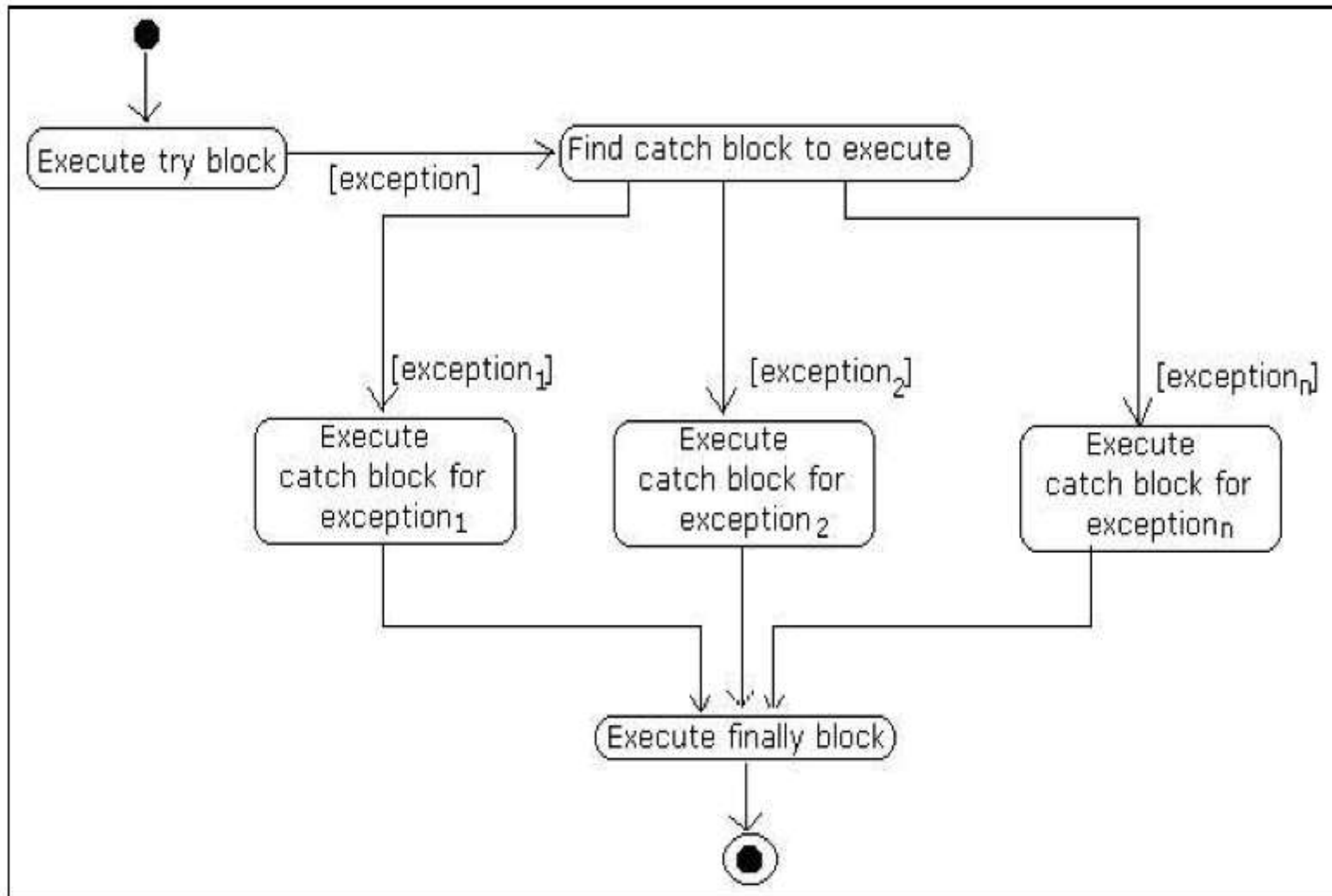


# try-catch-finally block

- Key aspects about the syntax of the try-catch-finally construct:
  - For each **try** block, there can be one or more **catch** blocks, but only one **finally** block.
  - The catch blocks and finally blocks must always appear in conjunction with the try block, and in the above order.
  - A try block must be followed by AT LEAST one catch block OR one finally block, or both.
  - Each catch block defines an exception handle. The header of the catch block takes exactly one argument, which is the exception its block is willing to handle. The exception must be of the Throwable class or one of its subclasses.



# Program Flow





# Example: try-catch

```
public class ExceptionExample {  
    public static void main( String[] args ){  
  
        try{  
            System.out.println( args[1] );  
        }  
        catch( ArrayIndexOutOfBoundsException exp ){  
            System.out.println("Exception caught!");  
        }  
    }  
}
```



# Summary

- Defined what exceptions are and some sample exceptions we encountered along the way.
- How to handle exceptions by using the try-catch-finally block.

