JYOTHISHMATHI INSTITUTE OF TECHNOLOGY AND SCIENCE

DEPARTMENT OF COMPUTER SCIENCE ENGINEEING

OBJECT OREINTED PROGRAMMIN THROUGHT JAVA

Exception Handling(UNIT-III)

V.NAREENKANTH

ASST. PROFESSOR

CSE DEPT

Exception Handling

- You learned that there are three categories of errors: syntax errors, runtime errors, and logic errors. *Syntax errors* arise because the rules of the language have not been followed. They are detected by the compiler. *Runtime errors* occur while the program is running if the environment detects an operation that is impossible to carry out. *Logic errors* occur when a program doesn't perform the way it was intended to.

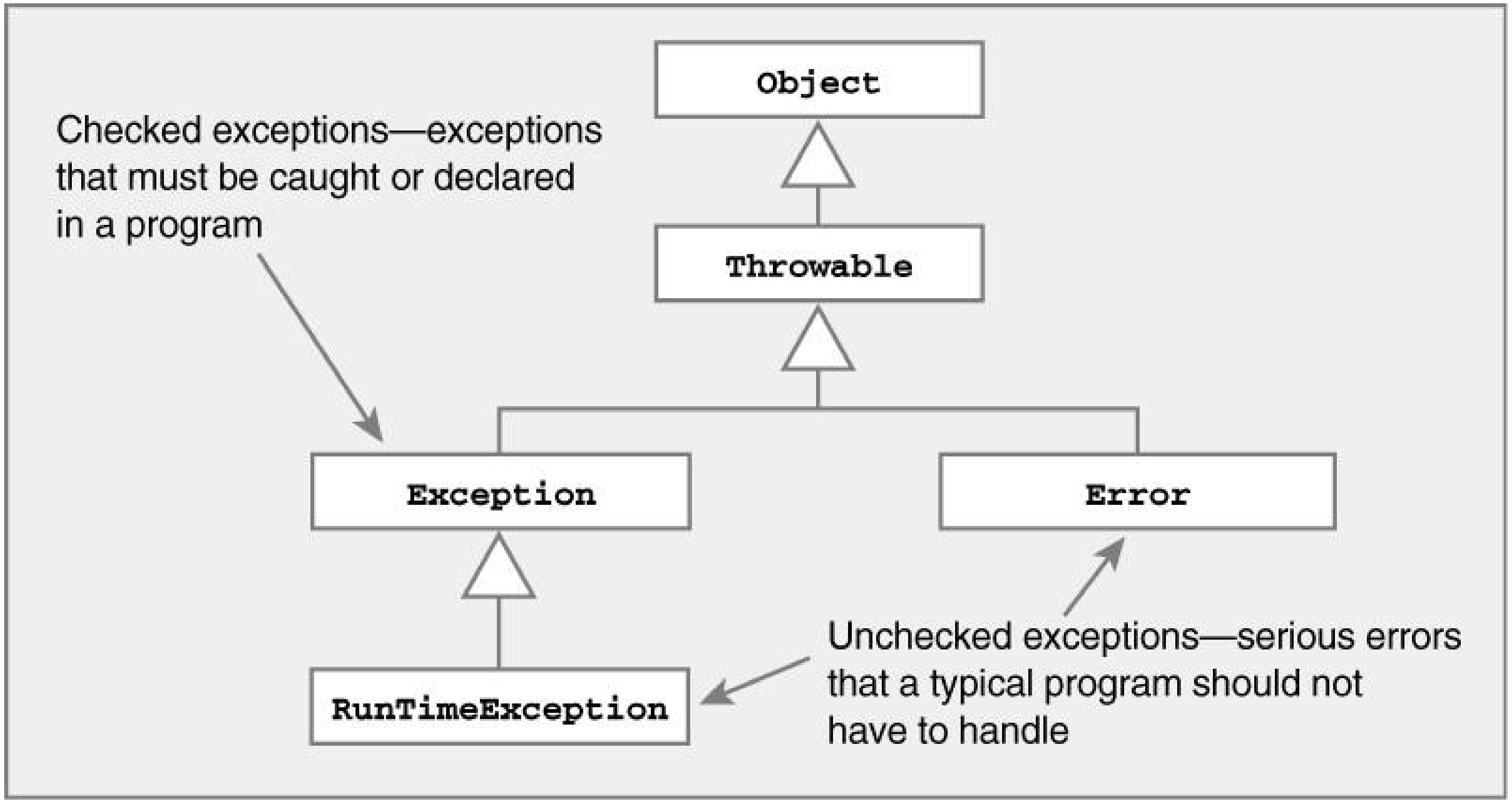
Exception Handling-Fundamentals

- ✓ An exception is an abnormal condition that arises in a code sequence at run time
- ✓ A Java exception is an object that describes an exceptional condition that has occurred in a piece of code
- ✓ When an exceptional condition arises, an object representing that exception is created and thrown in the method that caused the error
- ✓ An exception can be caught to handle it or pass it on
- ✓ Exceptions can be generated by the Java run-time system, or they can be manually generated by your code

Exception Handling

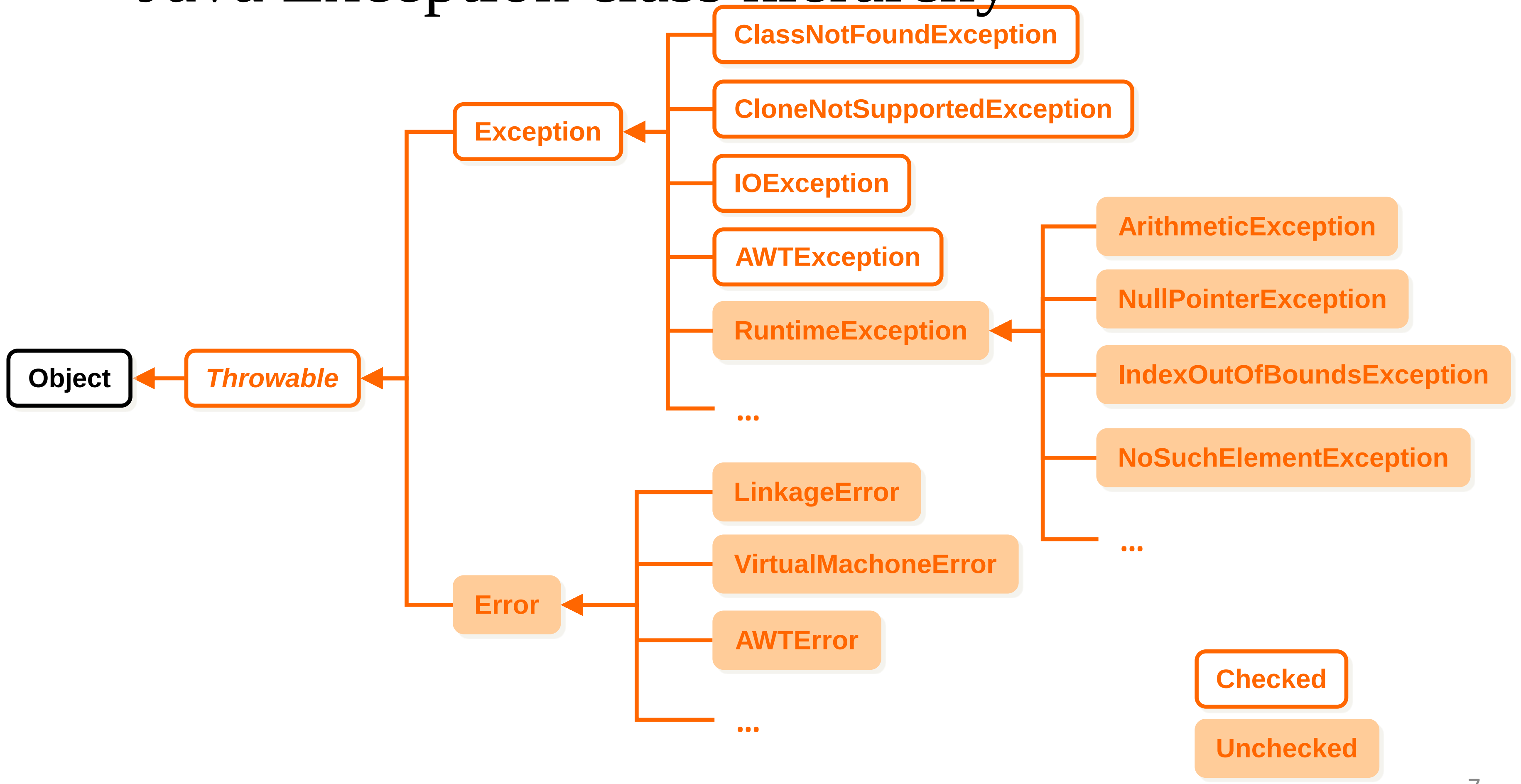
- ✓Performing action in response to exception
- ✓Examples
  - ✓Exit program (abort)
  - ✓Deal with exception and continue
    - ✓Print error message
    - ✓Request new data
    - ✓Retry action

Representing Exceptions


Object
Throwable
Exception
Error
RunTimeException
Checked exceptions—exceptions that must be caught or declared in a program
Unchecked exceptions—serious errors that a typical program should not have to handle

Representing Exceptions

✓ Java Exception class hierarchy



Checked / Unchecked

- RuntimeException, Error and their subclasses are known as *unchecked exceptions*. All other exceptions are known as *checked exceptions*, meaning that the compiler forces the programmer to check and deal with the exceptions.

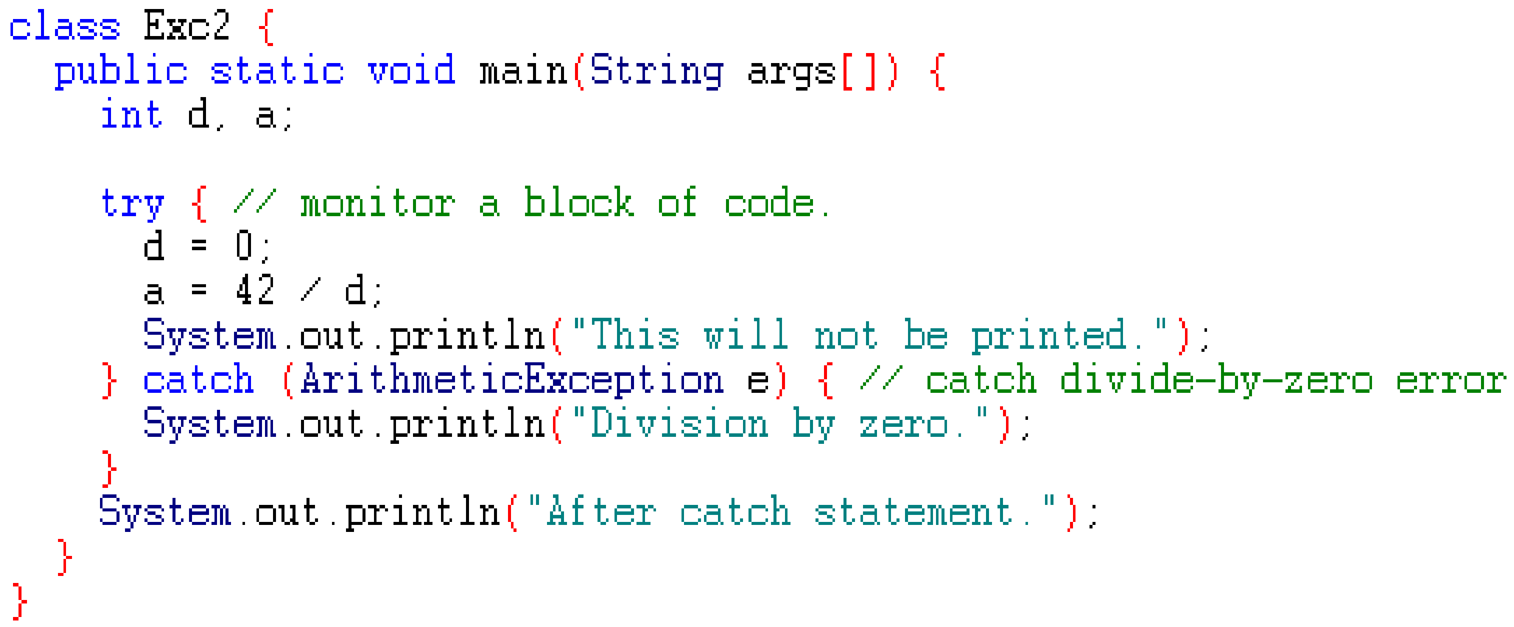
- In most cases, unchecked exceptions reflect programming logic errors that are not recoverable. For example, a NullPointerException is thrown if you access an object through a reference variable before an object is assigned to it; an IndexOutOfBoundsException is thrown if you access an element in an array outside the bounds of the array.

- These are the logic errors that should be corrected in the program. Unchecked exceptions can occur anywhere in the program.
- To avoid cumbersome overuse of try-catch blocks, Java does not mandate you to write code to catch unchecked exceptions.

Exception Handling in Java

- ✓ Java exception handling is managed by via five keywords: **try**, **catch**, **throw**, **throws**, and **finally**
- ✓ Program statements to monitor are contained within a **try** block
- ✓ If an exception occurs within the **try** block, it is thrown
- ✓ Code within **catch** block catch the exception and handle it

Example



```
class Exc2 {
  public static void main(String args[]) {
    int d, a;

    try { // monitor a block of code.
      d = 0;
      a = 42 / d;
      System.out.println("This will not be printed.");
    } catch (ArithmeticException e) { // catch divide-by-zero error
      System.out.println("Division by zero.");
    }
    System.out.println("After catch statement.");
  }
}
```

Output:

Division by zero.

After catch statement.