COMP90041

Programming and Software Development

Semester 1, 2021

Lab 9

Andrew Naughton

andrew.naughton@unimelb.edu.au

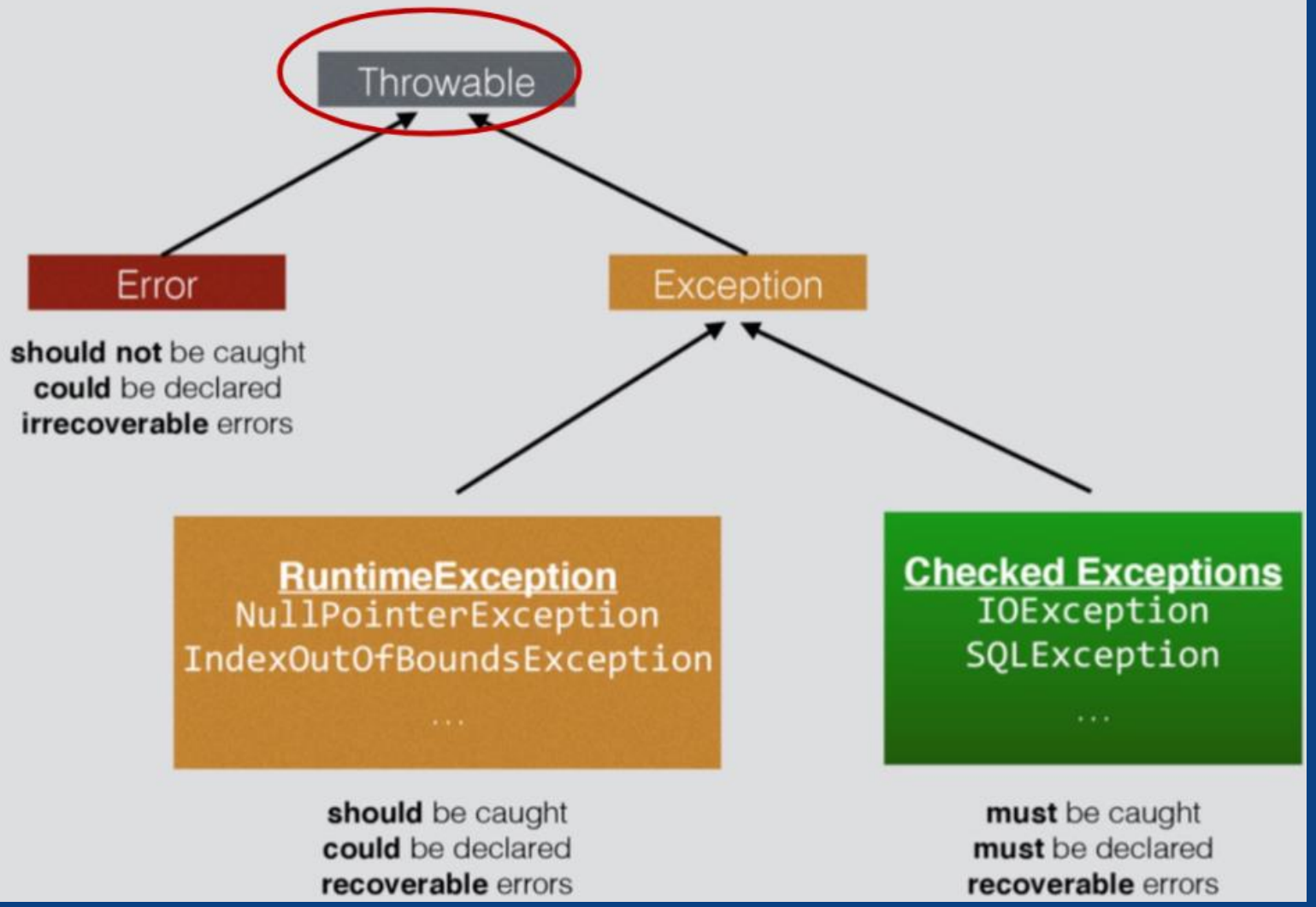
THE UNIVERSITY OF MELBOURNE

Outline

- Java Exception Hierarchy
- Exceptions
- throw
- Handling Exceptions
- catch
- finally
- Defining Exceptions

THE UNIVERSITY OF MELBOURNE

The Java Exception Hierarchy



Exception

- An object which tells us what went wrong

- Methods of all exception classes:
  - `toString()`: returns a String describing the exception
  - `getMessage()`: returns a String with detail about the error
  - `printStackTrace()`: prints a backtrace of what was happening (only useful to programmers)

THE UNIVERSITY OF MELBOURNE

throw

- Programs can throw exceptions
- This will interrupt the code that is currently executing
- If the exception is never caught (i.e. handled), the program aborts and a backtrace is printed

```
public Person(int age, String name){

    if (name == null){
        throw new NullPointerException("null name!");
    }
    this.name = name;
    this.age = age;
}
```

- Form:
- throw new <Exception Class>(<String detailing event>)

THE UNIVERSITY OF MELBOURNE

Handling Exceptions

```
try {
    code that may go wrong...
} catch (ExceptionClass var) {
    code to handle exception...
}
```

- `try`: specifies code that could throw an exception
- `catch clause`: specifies the kind of exception to catch
- `Inside catch`: specifies what to do if this exception occurs

THE UNIVERSITY OF MELBOURNE

catch

- `catch clause: catch(NullPointerException npe)`
- Can have multiple catch clauses. But note:
  - Only one handler is executed, the rest (if any) are ignored
  - First one that matches the thrown exception is used
- Always put more specific catches before general ones

THE UNIVERSITY OF MELBOURNE

What will this code print?

```
try {
    int i = 1;
    if (i > 0) throw new Exception();
    System.out.print("X");
} catch (Exception e) {
    System.out.print("Y");
}
System.out.println("Z");
```

A. X
B. XZ
C. Y
D. YZ
E. XYZ

catch ctd.

- By handling exceptions with catch, we recover from error
- Only exception(s) which are thrown inside the try can be caught in that try…catch

THE UNIVERSITY OF MELBOURNE

finally

- `finally` block is executed almost no matter what
- Finally is missed only if try or catch:
  - get stuck in an infinite loop; or
  - call `System.exit`

```
try {
    ...
} catch (...) {
    :
} finally {
    code to execute regardless
}
```

THE UNIVERSITY OF MELBOURNE

Define Exceptions

- Must inherit from Exception class, i.e.
  - `public class CustomException extends Exception`

- Usually define two constructors:
  - 1. no arguments and a default error message
  - 2. a single String argument (error message)

```
public MyException(String msg) { super(msg); }
public MyException() {
    super("default description string");
}
```

THE UNIVERSITY OF
MELBOURNE

Exercises