# Exceptions

3 Feb 2010
CMPT166
Dr. Sean Ho
Trinity Western University

## Exceptions for error handling

- Recall that exceptions are used for indicating runtime errors
  - Incorrect user input or parameters
  - No memory, disk space, permissions, etc.
- When an exception is thrown:
  - Execution of the current block is terminated
  - Search for the nearest exception handler
    - **Search enclosing blocks ({})**
    - **Search down the call-stack (what code invoked the current function)**

# Exceptions in Java

- In Java, use try-throw-catch
- Create an instance of java.lang.Exception and throw it:
  - **try {**
    - **if (s1.ID <= 0)**
      **throw new Exception("Invalid ID!");**
  - **} catch (Exception e) {**
    - **…**
  - **}**
- Can have several catch blocks, for different kinds of exceptions (first matching one is used)

# The caught exception object

- `} catch (Exception e) { …`
- A reference to the caught exception object is in e
  - Can use this to unpack auxiliary data
- The constructor for the Exception class may take a string argument: stored with the exception
  - `new Exception(“Invalid ID!”)`
- Get the string with the .getMessage() method on the caught exception object inside the handler:
  - `System.out.println( e.getMessage() );`

# Custom Exception classes

- Create your own type of exceptions:
    - `public class StudentError extends Exception`
- Need at least 2 constructors: no arg, 1 string arg
    - Pass the string msg up to superclass constr.:

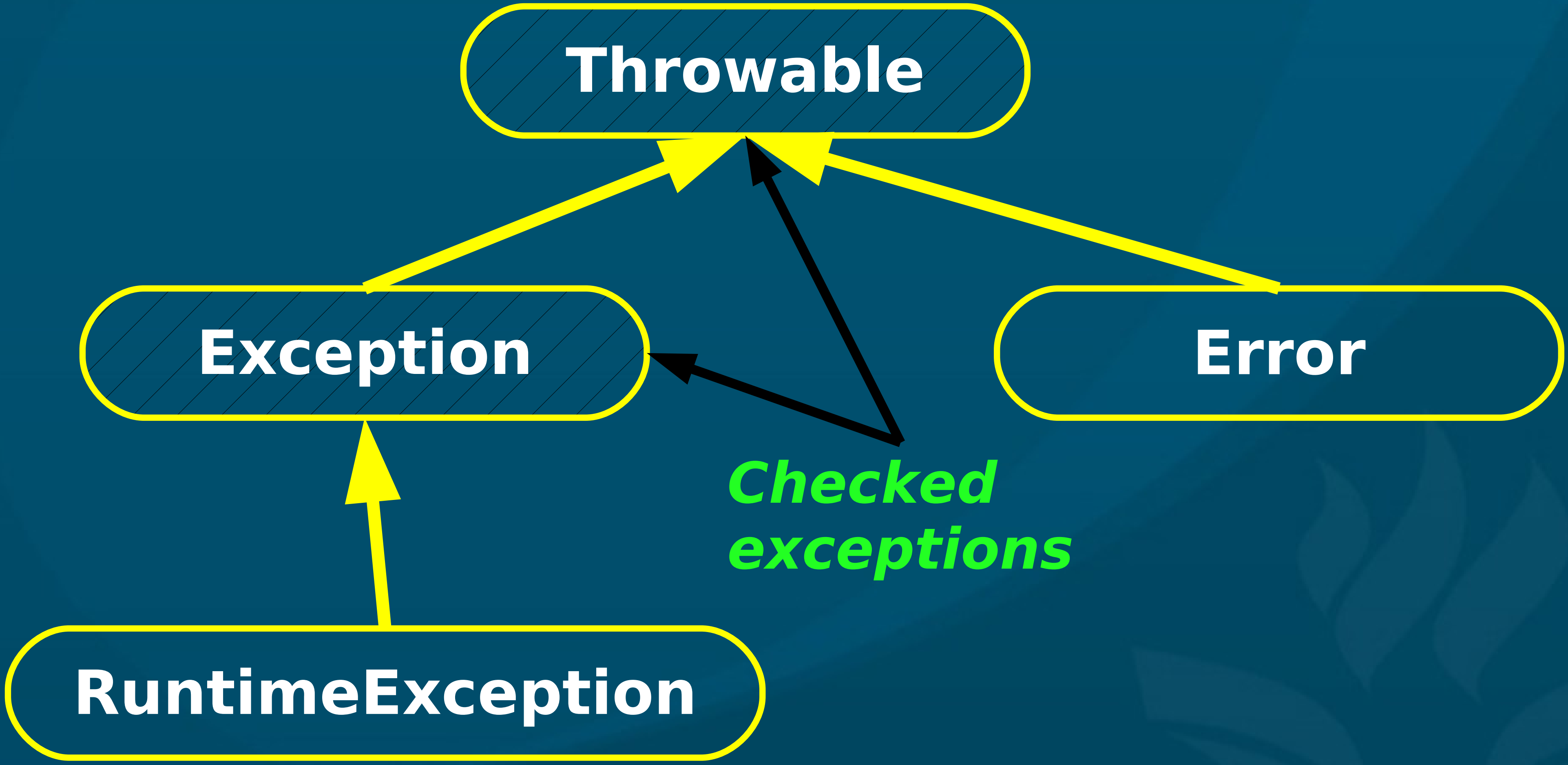
        -
          ```
          public StudentError( String msg )
            { super(msg); }
          ```

        -
          ```
          public StudentError()
            { super("Error with student!"); }
          ```
- Can also add your own auxiliary data (attributes) and constructors, set/get methods, etc.
    - `int studentID;`

## The catch-or-declare rule

- A method may encounter exceptions:
  - Directly thrown: throw new StudentError(…)
  - Or thrown by functions it calls: nextInt()
- For checked exceptions, the method must either:
  - Catch the exception and handle it, or
  - Declare that this method may raise an exception, and “pass the buck”:
    - **public void setID(int ID) throws StudentError { … }**

# Exception class hierarchy



## Exceptions raised by Scanner

- Using Scanner to read console input:
  - **import java.util.Scanner;**
  - **Scanner kbd = new Scanner(System.in);**
- Expecting an integer:
  - **int num = kbd.nextInt();**
- If Scanner can't convert the input to the desired type, it raises an InputMismatchException
- This can be caught, so you can try again