CISC 3115 TY2
Declaring and Throwing Exceptions

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Discussed
  - Approaches to handle errors (what-if and exceptions)
  - Concept of Exception
  - The Java throwable class hierarchy
    - system errors, runtime exceptions, checked errors, unchecked errors
- To discuss methods of
  - Declaring exception
  - Throwing exception,
  - Catching exception
  - Rethrowing exception

Motivation

- We can use exceptions to *separate notifying error from handling error*

- Is the following sufficient?

The Exception Approach: Advantage

- Separate notifying error from handling error

```
public static int quotient(int n1, int n2) {
    if (n2 == 0) {
        throw new ArithmeticException(
                "Divisor cannot be zero.");
    }
    return n1 / n2;
  }
}
```
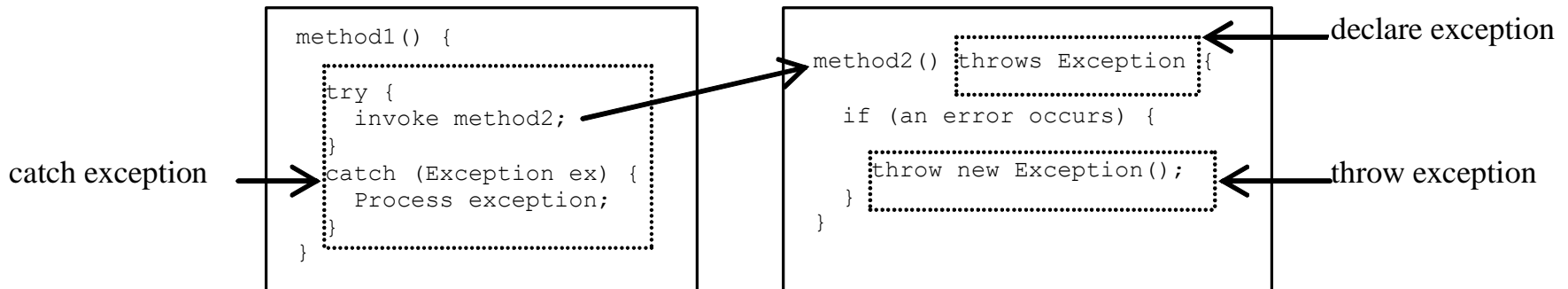
Notifying the caller an error occurred

```
public static void main(String[] args) {
                ……
    try {
      int result = quotient(n1, n2);
      System.out.println(n1 + " / " + n2 +
                " is "  + result);
    } catch (ArithmeticException e) {
      System.out.println("Exception: " +
            e.getMessage());
    }
```

Handling the error upon receiving the notification

The Big Picture

- Declaring, Throwing, and Catching Exceptions

```
method1() {

   try {
      invoke method2;
   }
   catch (Exception ex) {
      Process exception;

   }
}
```

```
method2() throws Exception {

   if (an error occurs) {

      throw new Exception();
   }
}
```

declare exception

throw exception

catch exception

Declaring Exception

- Every method must state the types of <u>checked exceptions</u> it might throw.

- Example

  public void myMethod() throws IOException { …

  }

- One may declare one or more exceptions to be thrown

- Example

  ublic void myMethod() throws IOException, OtherException { …

  }

Throwing Exceptions

- One can <u>create</u> an instance of an appropriate exception type and <u>throw</u> it in the method.

- Examples

    throw new TheException();

  Or

    TheException e = new TheException();
    throw e;

where TheException is a subclass of Throwable.

Declaring and Throwing Exceptions: Example

```
/** Set a new radius */
public void setRadius(double newRadius)
    throws IllegalArgumentException {
  if (newRadius >= 0) {
    radius =  newRadius;
  } else {
    throw new IllegalArgumentException(
      "Radius cannot be negative");
  }
```
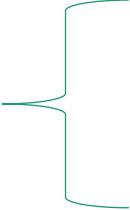
Declaring

Throwing

Catching Exception

- There are a few variations of try … catch …

- Frequently used:

    try { // Statements that may throw exceptions

    } catch (Exception1 exVar1) { handler for exception1;

    } catch (Exception2 exVar2) { handler for exception2;

    } … // more catch

    catch (ExceptionN exVarN) {

      handler for exceptionN;

    }

Questions

- Mechanism to declaring, throwing, and catching/handling exceptions

Exercise 1

- Following the lecture, design and implement a DivideByZero class where

  - Write a quotient method that takes two parameters and perform division of the two and state the exception it throws when the denominator is 0

  - Write another method that invokes the quotient method and catch and handle the exception the method throws

  - Make sure that you create directories for this exercise and submit it as part of the journal