

CISC 3115 EWQ6

# Throwing and Rethrowing 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
    - Methods of declaring, throwing, and catching exception
- To discuss methods of
  - Rethrowing exceptions

# Throwing Checked Exceptions

- When a method encounters a checked exception, the checked exception must be
  - declared to be thrown when declaring the method, or
  - be caught and handled.

# Declaring a Checked Exception to be Thrown

- IOException is a checked exception (how do we know?)
- Example: p2 throws an IOException, a checked exception

```
void p2() throws IOException {  
    if (a file does not exist) {  
        // e.g., using Files.exists(...)  
        throw new IOException("File does not exist");  
    }  
  
    ...  
}
```

# Caller p1() Must ...

- p2 throws an IOException, a checked exception
- p1 must catch it or throw it in its declaration.

```
void p1() {  
    try {  
        p2();  
    }  
    catch (IOException ex) {  
        ...  
    }  
}
```

(a)

```
void p1() throws IOException {  
    p2();  
}
```

(b)

# Questions

- Throwing checked exceptions

# What if we want ...

- p2 throws an IOException, a checked exception
- p1 must catch it or throw it in its declaration.

```
void p1() {  
    try {  
        p2();  
    }  
    catch (IOException ex) {  
  
        // do something and let  
        // the caller to do something  
        // else...  
    }  
}
```

```
void p1() throws IOException {  
  
    p2();  
  
}
```

# Rethrowing Exception

```
try {  
    statements;  
}  
catch(TheException ex) {  
    perform operations before exits  
    (e.g., undo what has been done here);  
    throw ex;  
}
```



# Questions?

- How to rethrow exceptions
- How to rethrow checked exceptions