Packages in Java

Introduction

- Java enables you to organize your classes into **folders** that are called **packages**.
- To create a package, simply include a **package** command as the first statement in a Java source file.
- Any classes declared within that file will belong to the specified package.
- This is the general form of the package statement:

```
package pkg;
```

pkg is the name of the package.

Importing Packages

In a Java source file, **import** statements occur immediately following the package statement (if it exists) and before any class definitions. This is the general form of the **import** statement:

```
import pkg1 [.pkg2].(classname | *);
```

Example:

```
package mypackage;      //create a package

import java.util.Date; //Use the Date class from the util package
import java.io.*;
```

java.lang package

- All the standard Java SE classes are included with Java begin with the name java.

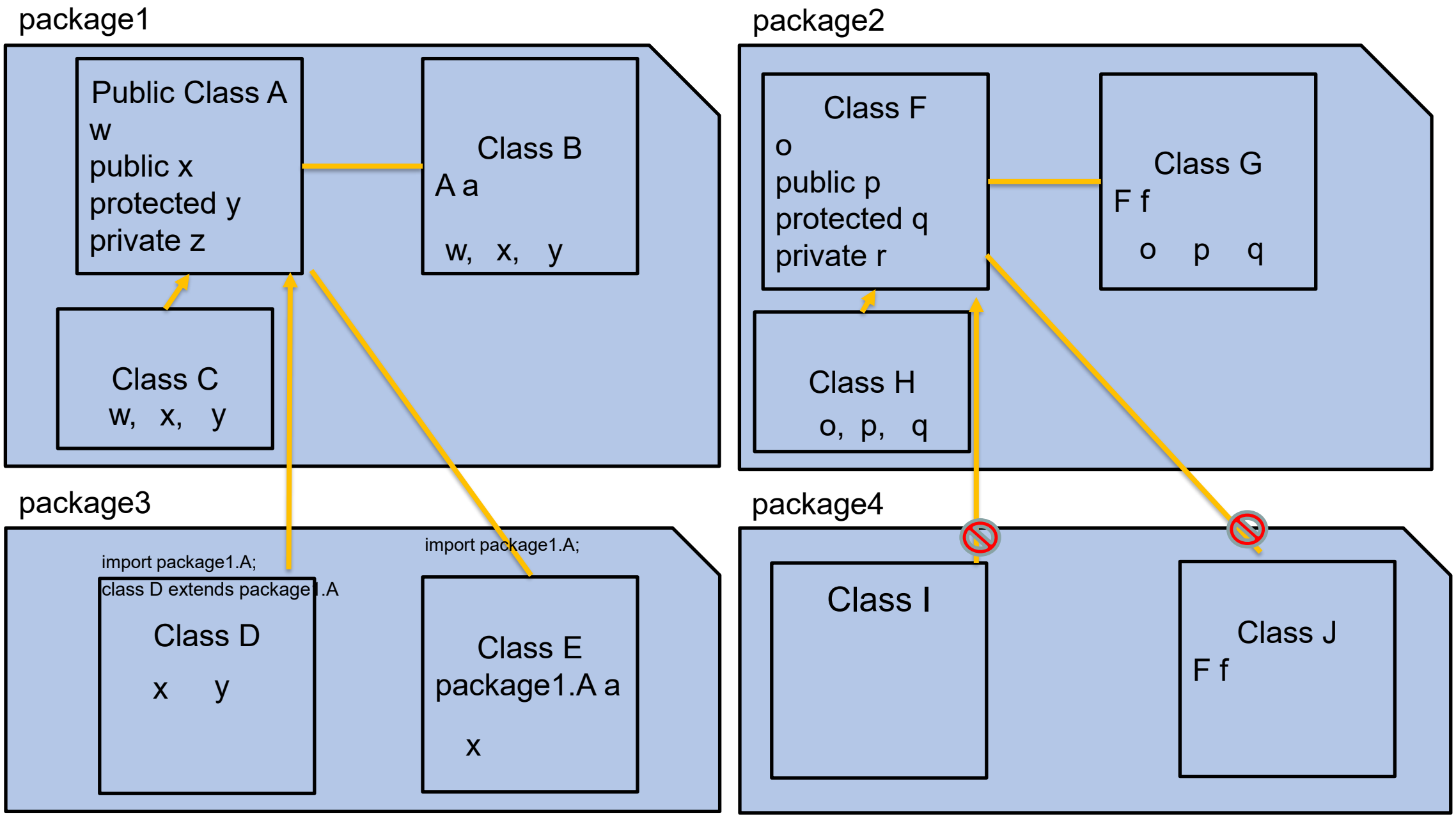
- The basic language functions are stored in a package called **java.lang**.

- Normally, you have to import every package or class that you want to use, but since Java is useless without much of the functionality in java.lang, it is implicitly imported by the compiler for all programs.

Modifiers & Access Specifiers

Modifiers and Access Specifiers cont'd

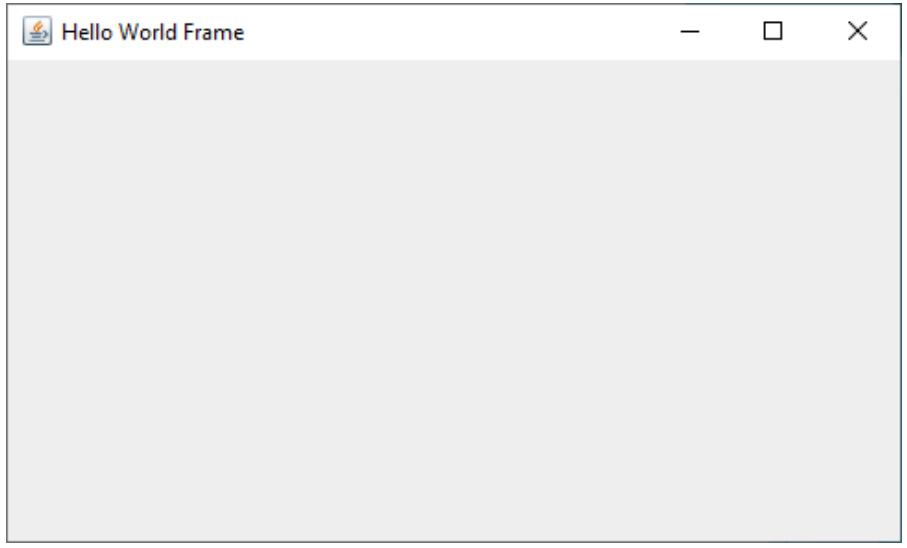
Keyword	Top Level Class	Methods	Variables	Free Floating Block
public	Yes	Yes	Yes	-
protected	-	Yes	Yes	-
(friendly)*	Yes	Yes	Yes	-
private	-	Yes	Yes	-
final	Yes	Yes	Yes	-
static	-	Yes	Yes	Yes
abstract	Yes	Yes	-	-
native	-	Yes	-	-
transient	-	-	Yes	-
volatile	-	-	Yes	-
synchronized	-	Yes	-	-


package1
Public Class A
w
public x
protected y
private z
Class B
A a
w, x, y
Class C
w, x, y
package2
Class F
o
public p
protected q
private r
Class G
F f
o p q
Class H
o, p, q
package3
import package1.A;
class D extends package1.A
Class D
x y
import package1.A;
Class E
package1.A a
x
package4
Class I
Class J
F f

Simple Java GUI

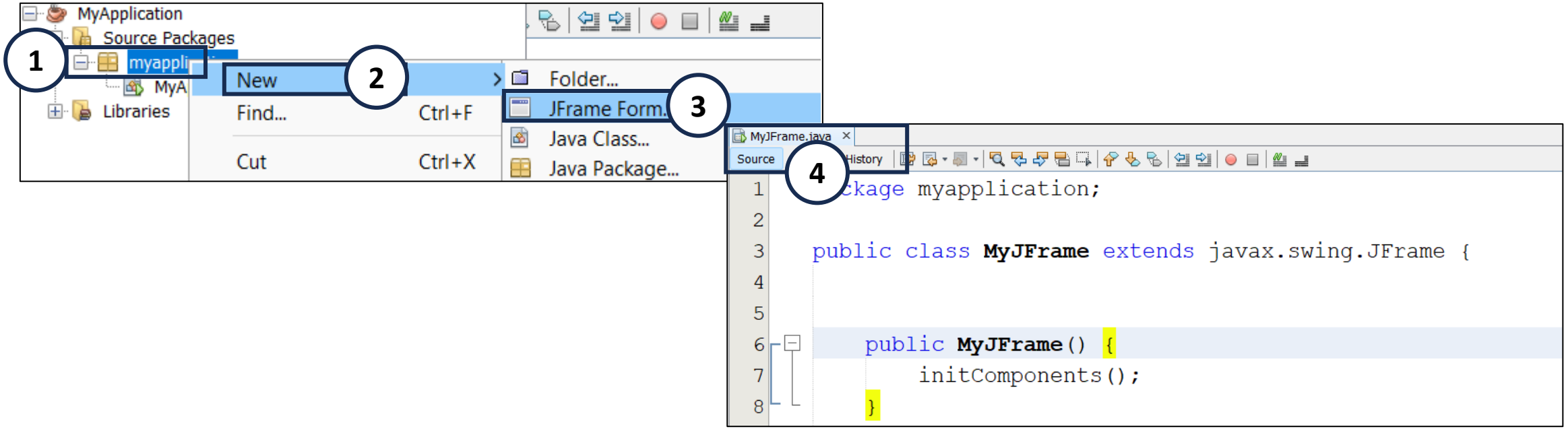
Create Simple GUI – java Swing Frame

- JFrame is a class of the javax.swing package .
- This is the top-level window, with border and a title bar.
- JFrame class has various methods which can be used to customize it.



Create Simple GUI – NetBeans

- After creating a new project on the NetBeans IDE,
  select the source package -> click new -> choose JFrame Form
- When created, open the source tab as shown in the picture, start overriding **`paint()`**



Create Simple GUI – Example

```
public class MyFrame extends javax.swing.JFrame {

    Date d;

    public MyFrame() {
        initComponents();
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        d = new Date();
        g.drawString(d.toString(), 100, 100);
    }
```

Graphics Class

- The Graphics object is your means of communication with the graphics display.
- You can use it to draw strings, basic shapes, and show images.
- You can also use it to specify the color and font you want.
- You can write a string using the following method:

```
void drawString(String str, int x, int y)
```

Graphics Class

- Some basic shapes can be drawn using the following methods:

```
void drawLine(int x1, int y1, int x2, int y2);

void drawRect(int x, int y, int width, int height);

void fillRect(int x, int y, int width, int height);

void drawOval(int x, int y, int width, int height);

void fillOval(int x, int y, int width, int height);

 void setColor(Color c);
```

Color Class

- There are 13 predefined color objects in Java.
- They are all declared as **`public static final`** objects in class Color :
  - **`Color.RED`**
  - **`Color.ORANGE`**
  - **`Color.PINK`**
  - **`Color.YELLOW`**
  - **`Color.GREEN`**
  - **`Color.BLUE`**
  - **`Color.CYAN`**
  - **`Color.MAGENTA`**
  - **`Color.GRAY`**
  - **`Color.DARK_GRAY`**
  - **`Color.LIGHT_GRAY`**
  - **`Color.WHITE`**
  - **`Color.BLACK`**

Exceptions

Exceptions

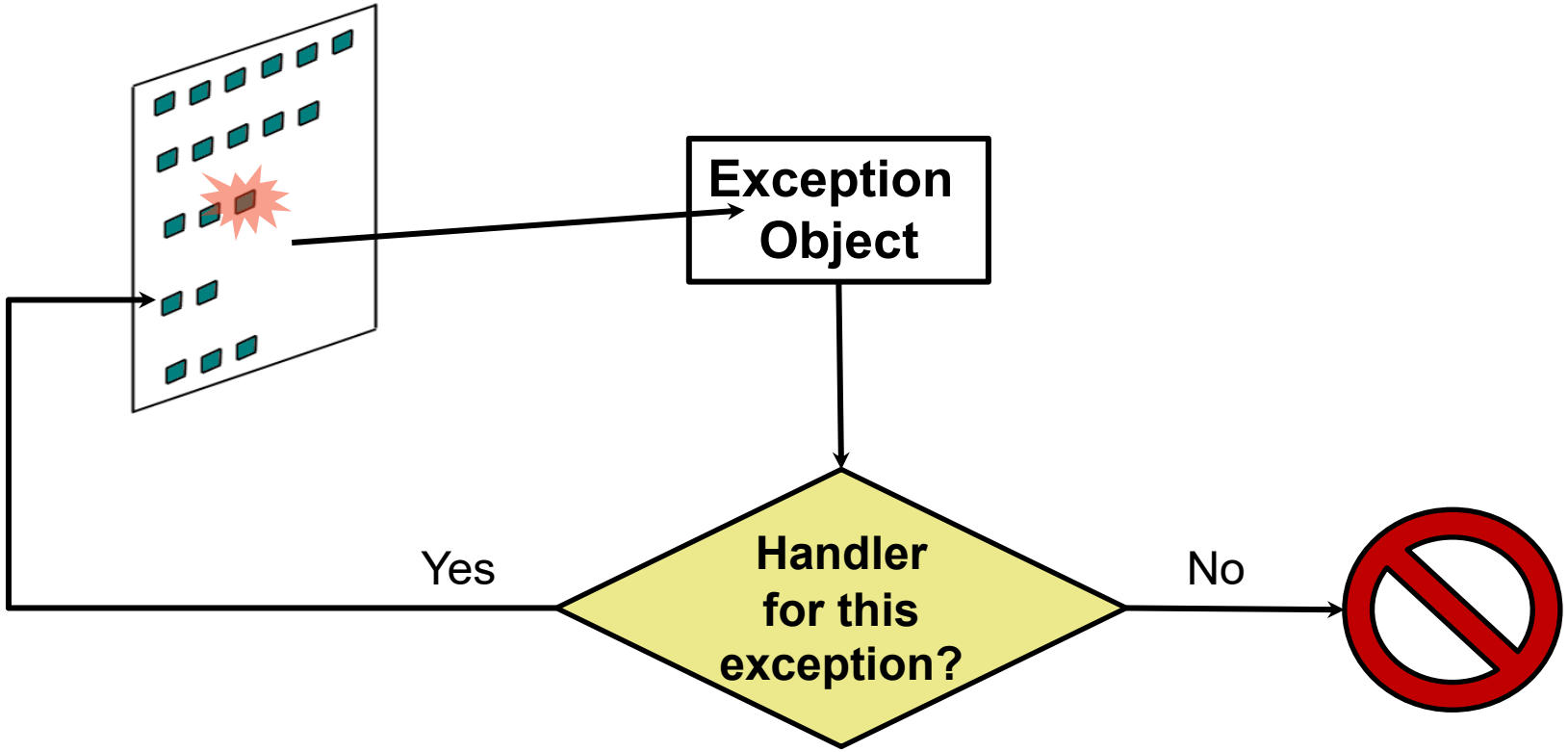
- An exception is an object that's created when an abnormal situation arises in your program **during runtime.**

- **Example:**
  - attempting to open a file that does not exist, or
  - attempting to write in a file that the OS has marked as read only.
  - attempting to use a reference whose value is null, or
  - attempting to access an array element that is beyond the actual size of the array.

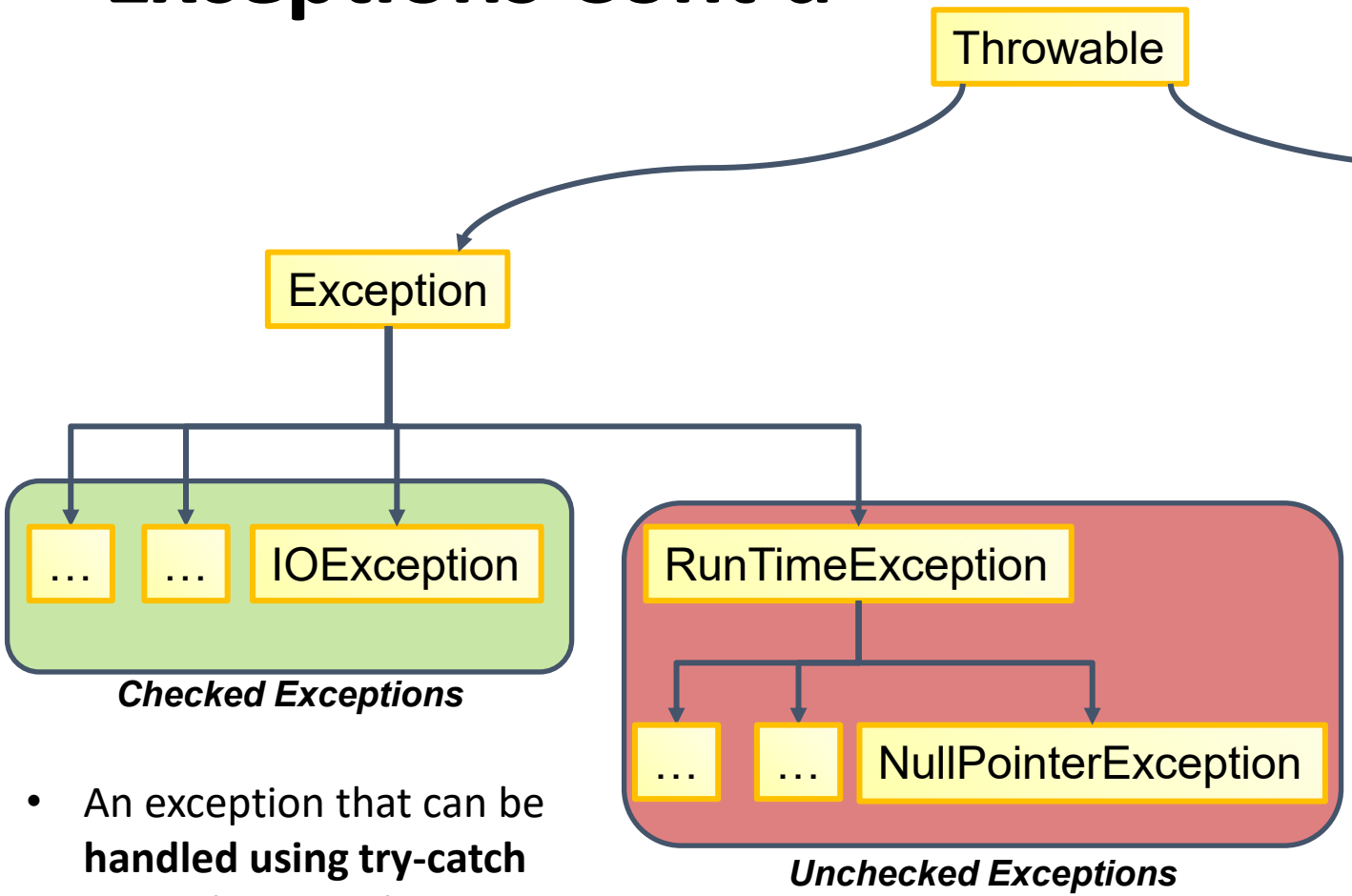
Exceptions Cont'd

- The exception object has description about the nature of the problem.
- The exception is said to be ***thrown*** when the problem occurs
- The code receiving the exception object as a parameter is said to ***catch it.***

How Does Java Handle Exceptions?



Exceptions Cont'd



- An exception that can be **handled using try-catch**
- Stops the compilation

- **Must NOT** be handled using try-catch; it's a logical problem that should be fixed
- Doesn't stop the compilation

- Can't be handled at all
- Terminates the program during runtime

Handling Exceptions

1. Including three kinds of code blocks to handle them:
   `try`, `catch`, and `finally`.

- **The `try` block:**

  encloses the code that may throw one or more exceptions.

- **The `catch` block:**

  encloses code that **handles exceptions** of a particular type that may be thrown in the associated **`try`** block.

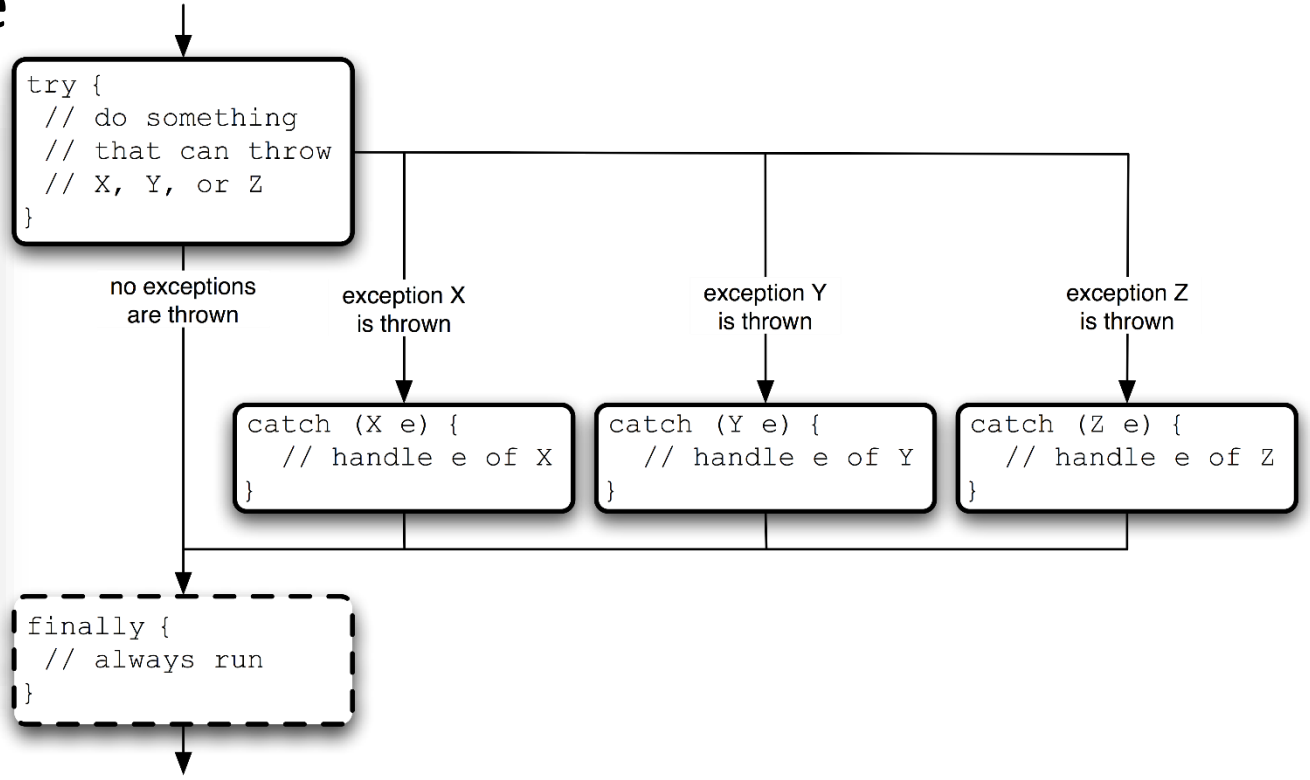
- **The `finally` block:**

  is used to write code that will **always definitely be executed** before the method ends, *whether the exception occurs or not.*

Handling Exceptions

- The try-catch-finally structure

```
try {
    // Code bock
}
catch (ExceptionType1 e1) {
    // Handle ExceptionType1 exceptions
}
catch (ExceptionType2 e2) {
    // Handle ExceptionType2 exceptions
}
// ...
finally {
    // Code always executed after the
    // try and any catch block
}
```



Considerations Regarding Exceptions

- If several method calls throw different exceptions,
  - then you can do either of the following:
    1. Write separate `try-catch` blocks for each method.
    2. Put them all inside the same `try` block and then

       write multiple `catch` blocks for it

       (one `catch` for each exception type).
    3. Put them all inside the same `try` block and then

       just `catch` the parent of all exceptions: `Exception`.

Considerations Regarding Exceptions

- If more than one `catch` block is written after each other,
  - Then you must take care not to handle a parent exception before a child exception (i.e., a parent should not mask over a child).
  - Anyway, the compiler will give an error if you attempt to do so.

Lab Exercises

Lab Assignment

Create a JFrame that makes use of the Graphics class drawing methods to draw the following lamp: