**About throw, throws and finally**

**throw**

The general form of **throw** is shown here:

throw *ThrowableInstance*;

Here, *ThrowableInstance* must be an object of type **Throwable** or a subclass of **Throwable**.

Primitive types, such as **int** or **char**, as well as non-**Throwable** classes, such as **String** and

**Object**, cannot be used as exceptions. There are two ways you can obtain a **Throwable**

object: using a parameter in a **catch** clause or creating one with the **new** operator.

Here is a sample program that creates and throws an exception. The handler that

catches the exception rethrows it to the outer handler.

**package** practice;

**class** MyClass {

**static** **void** demoproc() {

**try** {

**throw** **new** NullPointerException("demo");

} **catch**(NullPointerException e) {

System.*out*.println("Caught inside demoproc.");

**throw** e; // rethrow the exception

}

}

**public** **static** **void** main(String args[]) {

**try** {

*demoproc*();

} **catch**(NullPointerException e) {

System.*out*.println("Recaught: " + e);

}

}

}

**throws**

A **throws** clause lists the types

of exceptions that a method might throw.

This is the general form of a method declaration that includes a **throws** clause:

*type method-name*(*parameter-list*) throws *exception-list*{

// body of method

}

Here, *exception-list* is a comma-separated list of the exceptions that a method can throw.

**finally**

**finally** creates a block of code that will be executed after a **try /catch** block has completed

and before the code following the **try/catch** block. The **finally** block will execute whether

or not an exception is thrown. If an exception is thrown, the **finally** block will execute even

if no **catch** statement matches the exception.

The **finally** clause is optional. However, each **try** statement requires at least

one **catch** or a **finally** clause.

**package** practice;

// Demonstrate finally.

**class** MyClass {

// Throw an exception out of the method.

**static** **void** procA() {

**try** {

System.*out*.println("inside procA");

**throw** **new** RuntimeException("demo");

} **finally** {

System.*out*.println("procA's finally");

}

}

// Return from within a try block.

**static** **void** procB() {

**try** {

System.*out*.println("inside procB");

**return**;

} **finally** {

System.*out*.println("procB's finally");

}

}

// Execute a try block normally.

**static** **void** procC() {

**try** {

System.*out*.println("inside procC");

} **finally** {

System.*out*.println("procC's finally");

}

}

**public** **static** **void** main(String args[]) {

**try** {

*procA*();

} **catch** (Exception e) {

System.*out*.println("Exception caught");

}

*procB*();

*procC*();

}

}

**Types of exceptions:**



