# Lab CC 3: Handling Exceptions

## Objectives

In this lab you will see the effect that throwing an exception has in a thread.

## Instructions

### Step 1 Create the Thread class

1. Use the same Task class from the previous lab which implements the Runnable interface.
2. Use the same Runner class as in the previous lab.

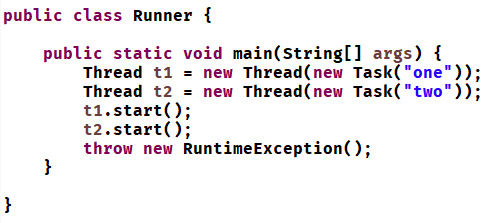
### Step 2: Throw an exception in one of the Threads

1. In the code for the Task class, add a line to throw an exception
2. Text

   Description automatically generatedUse a RuntimeException since this is an unchecked exception and we don’t need to catch it.
3. Text

   Description automatically generatedNotice that thread one stops and does not process iterations 6 through 10 while thread two continues to work.

### Step 3: Throw an exception in the main thread

1. Remove the line you added in step 2 so that the Task class does not throw an exception.
2. Add the following line into the runner class to throw a RuntimeException in the main thread
3. Run the code and notice that even through the main thread throws an exception, the two threads execute to completion.
4. In order to see this you may need to increase the sleep timeout so that the messages interleave properly

Text

Description automatically generated