# Lab CC 4: Executor Interface

## Objectives

In this lab you will implement a simple Executor Service using a fixed pool size Executor factory and then a customizable with the customizable class

## Part One Instructions

### Step 1 Create the Thread class

1. For this lab, we will create a runnable object that prints out information about itself
2. Make the class public and put it in its own file.
3. The example uses the package “pools”
4. The class implements Runnable but also prints out the name of the thread it is running in using a Thread static method.
5. Text

   Description automatically generatedA sleep method lets you slow down the output.

### Step 2: Create the Executor Service

1. For this part of the lab, use the static factory method
2. Text

   Description automatically generatedSubmit a series of tasks to the service

### Step 3: Run the Code

1. This code is non-deterministic – many different factors in your environment will affect how the threads are executed including how man and in what order
2. Experiment by increasing the sleep value and number of threads created

Graphical user interface, text, application

Description automatically generated

## Part Two Instructions

### Step 1: Modify the Executor Service code

1. You can just use a different runner class and keep the Task class from the previous part.
2. Graphical user interface, text, application

   Description automatically generatedCreate the Executor Service using the following code:

### Step 2: Run the code

1. No other changes have to be made to run the code

Graphical user interface, text

Description automatically generated

## Part Three: Instructions

In this section, you will experiment with tuning the initial values of the Executor Interface you created in the last section.

1. Set the size of the queue to 1.
2. Change the number of tasks submitted to 100
3. When you execute the application, you should get an exception that is thrown because there isn’t enough room in the queue to hold all the submitted jobs. One of the options available is to provide your own handler in order to manage this exception
4. Change the size of the pool back to 100 and now change the core size from 3 to 0. Run the code and notice that we still have a thread because the executor can scale up to five threads.