



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Parallel Programming
Assignment 1: Introduction to Parallel Programming
Spring Semester 2024

Assigned on: **19.02.2024**

Due by: **(Wednesday Exercise) 26.02.2024**
(Friday Exercise) 28.02.2024

Overview

The purpose of this exercise is to introduce you to the development environment that you are going to use for this course and to start learning to use the Java programming language.

In particular, in this exercise you will:

- install the Eclipse IDE, a multi purpose development environment that will be used throughout this course;
- create a simple project in the Eclipse IDE;
- learn how the exercise submission system for this course works;
- write a Java class;

This exercise has become quite comprehensive because it also comprises all details required to set-up your development environment and submit exercises. So, don't be scared, it is actually not that bad!

Exercise 1 – Set-up

The Eclipse IDE is one of the most popular integrated development environments (IDEs) for Java programming. Its purpose is to make the development process for a Java programmer as easy as possible. Although the variety of features Eclipse offers might seem overly complex at the beginning, especially for just writing simple programs, you will soon much appreciate the features the IDE has to support you during programming.

In this section we will give several references for getting started with Eclipse and Java. The steps you should follow are the following:

- **Setup Java** The easiest way is to navigate to <https://adoptopenjdk.net/> and installing the latest LTS release (jdk-17.0.6+ 10).
- **Setting up Eclipse** To do this, go to the following link and download the latest version of Eclipse for your operating system:
<https://www.eclipse.org/downloads/packages/release/2022-12/r/eclipse-ide-java-developers>
- **Start Eclipse** At startup, Eclipse asks for the directory where all your data will be stored (workspace).

- **Create an empty Java Project** Inside Eclipse, select the menu item File > New > Java Project. Select the project name to be **assignment1**. Then click Finish. If you are asked, don't create a module-info.java.
- **Create a Java Class** Your class should be named **HelloWorld**, the package should be named **assignment1**. To create a Java Class right-click on a project in the Package Explorer and select "New" and then "Class".
- **Hello World Class** Copy/Paste the code below into your new class

File: HelloWorld.java

```
package assignment1;

/* This is a class */
public class HelloWorld {

    /*
     * This is the main method.
     * This is the entry point for your program.
     */
    public static void main(String[] arguments) {

        /* Output Hello World to the console */
        System.out.println("Hello World");
    }
}
```

- **Running your program** In order to run your new Java Application with Eclipse, right click the "HelloWorld" class from the package explorer, click "Run As" and select "Java Application".

Deliverables: You need to submit this exercise as described in the Submission Section.

Submission

You need to submit your code and reports by using our submission system. This step will be the same for all of the subsequent exercises, but we will explain it in detail here so you can familiarize yourself with the system.

- **Set-up your project for check-in.**
 - Right click your created project called **assignment1**.
The name of the subfolder (i.e., your project name in Eclipse) - **assignment1** - is important for the evaluation of your exercise by your teaching assistant. In case your project is not called assignment1, right-click on the project, go to Refactor, then click **Rename** and rename it to **assignment1**.
 - In the menu go to **Team**, then click **Share Project**.
 - (Optional) If you are using git for the first time you may see a dialog "Identify Yourself" that asks for your name and email address.
 - You should see a dialog "Configure Git Repository". Here, next to the Repository input field click on **Create...**
 - Select a directory (different than Eclipse workspace) where you would like to store the assignments. Note for all your assignments you should use the same directory.
 - Click **Finish**.

- **Commit changes in your project**

- Now that you created an empty git repository, you need to make sure that every time you change your code or your report, at the end you commit your changes and send (push) them to the git server.
- Right click your project called **assignment1**.
- In the menu go to **Team**, then click **Commit...**
- In the Comment field, enter a comment that summarizes your changes.
- In the "Unstaged Changes" list, select all the files that you changed and want them to be committed. This typically includes all the Java files but not necessarily all the files (e.g., you don't have to commit setting files of our eclipse installation). Add the files after which you should see them in the "Staged Changes" list.
- Then, click on **Commit** to store the changes locally or **Commit and Push** to also upload them to the server. Note that in order to submit your solution you need to **both** commit and push your changes to the server.

- **Push changes to the git server**

- Right click your project called **assignment1**.
- In the menu go to **Team**, then click **Push Branch 'master'**. Note if this is not your first push you can also use **Push to Upstream** to speed up the process.
- A new dialog appears, now fill in for the URI field:
`https://gitlab.inf.ethz.ch/COURSE-PPROG24/pprog-<nethz-username>.git`
- Click **Preview**
- An authentication dialog should appear. Fill in your nethz username and password and click **Log in**.
- Click **Push** to confirm your changes. Note that eclipse might ask for authentication again.

- **Browse your repository online**

- You can access and browse the files in your repository online on GitLab at:
`https://gitlab.inf.ethz.ch/COURSE-PPROG24/pprog-<nethz-username>`

In rare cases, you might get an error like this:

Can't connect to any URI

`https://gitlab.inf.ethz.ch/COURSE-PPROG24/pprog-<nethz-username>.git`

`(https://gitlab.inf.ethz.ch/COURSE-PPROG24/pprog-<nethz-username>.git:`

`https://gitlab.inf.ethz.ch/COURSE-PPROG24/pprog-<nethz-username>.git/info/refs?`
`service=git-receive-pack not found)`

It means that no repository was created for your username yet. If that is the case, please contact your teaching assistant. The teaching assistant will be able to create the repository location for you. After that, you should be able to submit your work by following the instructions above.