

# EECS2101 (E) Summer 2025

## Assignment 1

### Programming with Recursion

Adapted from Prof. Jackie Wang's Assignment 1 and Submission Instructions for EECS2101

Instructor: Sunila Akbar

**Released: Saturday, May 10**  
**Due Date: 11:59 pm, Saturday, May 24**

- The format of this assignment closely reflects that of all subsequent assignments and programming tests. Please read the instructions carefully.
- Although group work is permitted, you are strongly encouraged to first attempt the assignment independently before seeking assistance.
- For this assignment you are provided with only a limited number of starter JUnit tests to help you begin. You are encouraged not only to use these starter tests, but also to supplement them with additional tests that evaluate your code across a wider range of input values. This practice will be particularly helpful in programming tests, where additional test cases beyond the provided ones may be used for grading.

For this assignment, make your best effort before the submission deadline. After submission, additional tests will be provided so that you can independently evaluate the robustness of your solution.

- Follow the instructions to submit (via web submit) the required file (a Java project archive zip file).
- Emailing your submission to the instructor or TAs will **not** be acceptable, under any circumstance.

- **Texts in blue** are hyperlinks to the corresponding documents/recordings.

## Policies

- **Please note that your solution to this assignment - whether submitted or not - remains the intellectual property of the EECS Department.** As such, we kindly ask that you do **not** share or distribute your code through any public platform (e.g., a non-private GitHub repository). Sharing solutions publicly may violate academic integrity policies, and the department reserves the right to take appropriate action if necessary.
- You are responsible for submitting your work electronically through the web submit system before the deadline. We strongly recommend that you **back up your work regularly** to prevent loss due to unexpected technical issues. **To help you manage your projects securely, we encourage you to follow this tutorial series on setting up a private GitHub repository for your Java code.**
- Please be aware that the deadline is **firm**. To ensure fairness for everyone, late submissions cannot be accepted. We recommend planning ahead and starting early to avoid last-minute issues.

# Contents

<b>0 Task 0: <u>Optional</u> Background Studies</b>	<b>4</b>
<b>1 Task 1: Complete Programming Tasks</b>	<b>5</b>
1.1 Step 1: Download and Import the Starter Project . . . . .	5
1.2 Step 2: Programming Tasks . . . . .	6
1.3 Step 3: Exporting the Completed Project . . . . .	7
<b>2 Submission</b>	<b>8</b>
<b>3 Amendments</b>	<b>9</b>

## Learning Outcomes

By completing the assigned exercises of this assignment, you are expected to be able to:

1. Catch up with the basics of recursion (by referring to the optional background study materials)
2. Implement and test recursive algorithms on integers and arrays.
3. Use the debugger tool to identify and fix errors.

## Assumptions

- You have already setup a Github account and stored work in a **private** repository: e.g., `EECS2101E-S25-workspace`.  
**Note.** Though not required, it is highly recommended that you adapt to the practice of backing your work using a versioning tool like Github.
- You are able to use Eclipse to complete this assignment on either your own machine or the EECS remote labs.  
**Note.** The starter project was created using Eclipse and an Eclipse project archive file is expected to be submitted. Therefore, you may not want to use other IDE such as IntelliJ.

## Requirements of this Assignment

In the context of a **programming test**:

- During the programming test:
  - You will be required to work on the Eclipse IDE. Otherwise the exported project from another IDE may fail the required structure (by the grading program) and receive a penalty.
  - If your submitted project does not compile (i.e., syntax errors or type errors) with **the given starter test class**, TAs will attempt to fix them (if they are quick to fix); once the revised submission is graded, your submission will receive a **penalty** on the resulting marks.
  - The grading of your submission will start by automatically **unzipping** the submitted Java project archive file (**.zip**) and extracting the required class(es). It is therefore crucial for you to follow **precisely** the spelling of the archive file name. **Penalty** will be taken if the grading cannot proceed due to carelessness on following the instructions in Section 1.3.
  - You will be graded not only by the starter JUnit tests given to you, **but also additional tests** covering some other input values. This is to encourage you to take more responsibility for the correctness of your code, by writing your own tests.

It is therefore your best interest of completing the assignment using the Eclipse IDE and practicing according to the above-mentioned expectations.

- For the JUnit test class `StarterTests.java` given to you:
  - Do **not** modify the test methods given to you.
  - If you wish to add new test methods, do so by creating another test class in the `tests` package.
- Derived from the given JUnit test methods:
  - Each class you introduce and implement must be placed under the `model` package.
  - Do **not** add any class that is not required by the starter tests: any such class will not be graded.
  - If considered necessary, you may declare additional attributes and/or helper methods if you wish.
  - For each method you implement:
    - \* No `System.out.println` statements should appear in it.
    - \* No Scanner operations (e.g., `input.nextInt()`) should appear in it.

Instead, declare input parameters of the method as indicated by the JUnit tests.

## 0 Task 0: Optional Background Studies

1. This assignment assumes the basics of recursive thinking and programming, as covered in the prerequisite course (EECS2030). If you're not yet confident with these concepts, the following resources - **courtesy of Prof. Jackie Wang** - may be helpful:
  - From [EECS2030 F'21 Lectures](#): Parts A - C, Lecture 8, Week 12
  - From [EECS2030 F'24 Lectures](#): Lecture 24, Sec. E (Tower of Hanoi)  
[ Slides: [here](#) ]
  - Recursion Lab from EECS2030 F'22: Starter project [here](#) [ Solution: [here](#) ]
  - Extra recursion problems: Problem descriptions [here](#); Starter project [here](#)  
[ Solution: [here](#); Tutorial recording: [here](#) ]
2. In addition to primitive arrays ([ ]), you will be asked to implement solutions using Java's standard collection framework:
  - [ArrayList](#) and [HashMap](#) [ e.g., [Quick tutorial by Prof. Jackie Wang](#) ]
3. This assignment also requires you to infer, from the given starter JUnit test methods, the required class(es) and method(s) to implement. If needed, please refer to this short guide prepared by Prof. Jackie Wang:

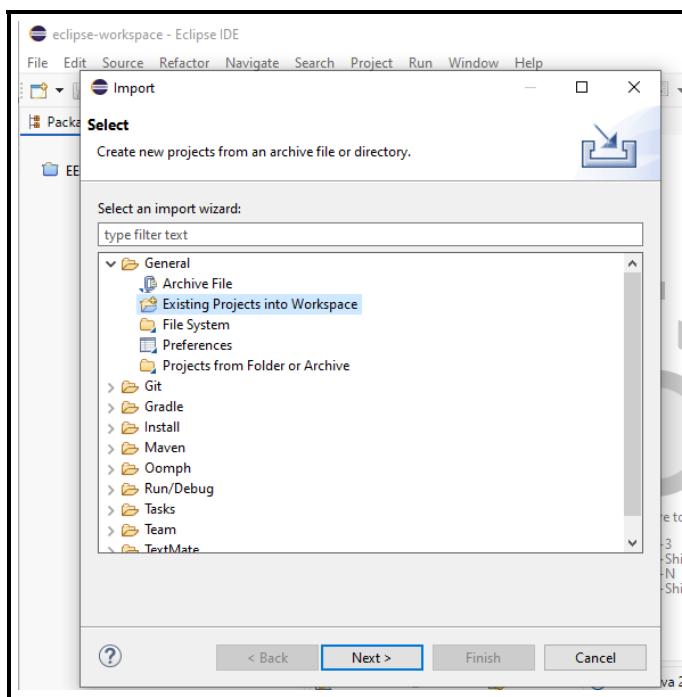
[https://www.eecs.yorku.ca/~jackie/teaching/lectures/2024/F/EECS2030/notes/EECS2030\\_F24\\_Inferring\\_Classes\\_from\\_JUnit.pdf](https://www.eecs.yorku.ca/~jackie/teaching/lectures/2024/F/EECS2030/notes/EECS2030_F24_Inferring_Classes_from_JUnit.pdf)
4. When you encounter issues implementing methods - especially during programming tests - the use of a **debugger** is strongly encouraged:
  - Debugging actions (Step Over, Step Into, Step Return) [ Parts C-E, Week 2 of [Java from Scratch \(W'21\)](#), [Prof. Jackie Wang](#) ]

# 1 Task 1: Complete Programming Tasks

Starting Task 1 should mean that you have *already completed* the background studies (if necessary) as outline in Section 0.

## 1.1 Step 1: Download and Import the Starter Project

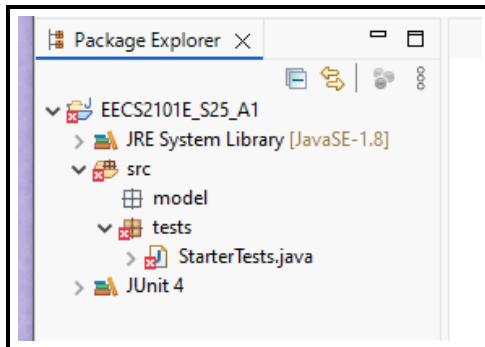
1. Download the Eclipse Java project archive file from eClass: **EECS2101E\_S25\_A1.Starter.zip**
2. Launch Eclipse and browse to, e.g., **EECS2101E-S25-workspace**, as the **Workspace** then click on **Launch**.
3. In Eclipse:
  - 3.1 Choose **File**, then **Import**.
  - 3.2 Under **General**, choose **Existing Projects into Workspace**.



- 3.3 Choose **Select archive file**. Make sure that the **EECS2101E\_S25\_A1** box is checked under **Projects**.
- 3.4 Then click **Finish**.

## 1.2 Step 2: Programming Tasks

From the **Package Explorer** of Eclipse, your imported project has the following structure:



- The **tests** package contains the **StarterTests** JUnit test class.
  - It is **expected** that the **StarterTests** JUnit class contains **compilation errors** to start with. This is because that declarations and definitions of the required class(es) and method(s) it references are missing.
  - Study carefully the test methods listed in this test class, as they suggest:
    - \* the required class(es) and method(s) to be implement in the ‘model’ package
    - \* how the required class(es) and method(s) should be implemented.
- You must **not** modify these given JUnit tests, as they suggest how the intended class(es) and method(s) should be declared.
- Test methods included here are meant to get you started. **Therefore, you are expected to write additional tests to ensure that your submitted code is able to handle other input values implied by the problem specification (see the in-line comments in StarterTests).**
- The **model** package is empty. Class(es) and method(s) derived from the given JUnit class **must** be added to this package. Class(es) added to a package other than **model** will **not** be graded.

Therefore, your tasks are:

1. Inferring from the given JUnit tests, add the missing class(es) and method(s) into the **model** package. For example, if you add class **Foo** in the model package, make sure that you write a line in the beginning of the **StarterTests** class (after the line `package tests;`):

```
import model.Foo;
```

2. Pass **all** JUnit tests given to you (i.e., a **green bar**).

To run them, as shown in the Review Tutorial Series, right click on **StarterTests.java** and run it as JUnit tests. Of course, none of the given tests would pass to begin with.

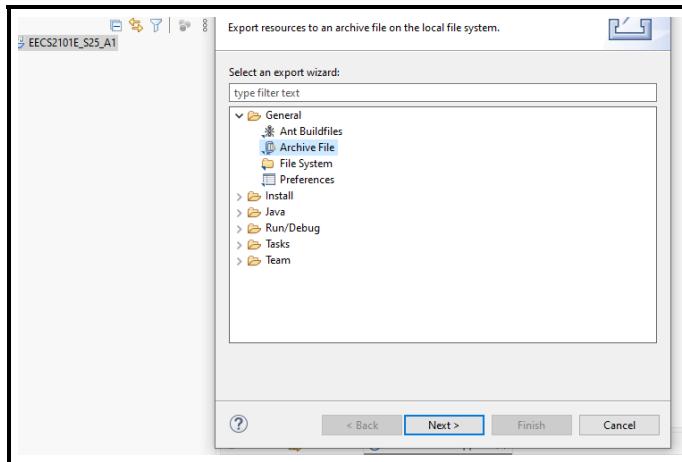
**How to Deal with a Failed JUnit Test?** From the JUnit panel from Eclipse, click on the failed test, then **double click** on the first line underneath **Failure Trace**, then you can see the **expected value** versus the **return value** from your implemented method. Furthermore, when needed, you should a **breakpoint** at the line of the failing assertion, then launch the **debugger** to pinpoint where the error came from.

### 1.3 Step 3: Exporting the Completed Project

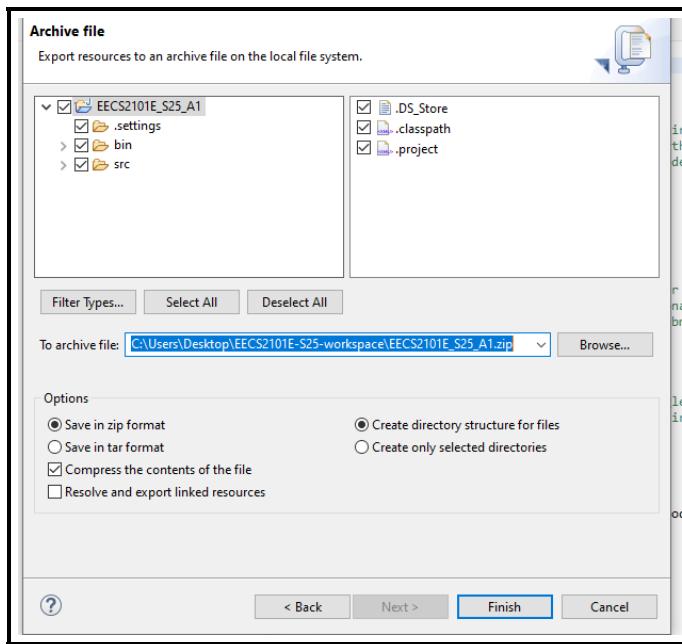
You are required to submit a Java project archive file (.zip) consisting all subfolders.

In Eclipse:

1. Right click on project EECS2101E\_S25\_A1. Then click **Export**.
2. Under **General**, choose **Archive File**.



3. Check the top-level EECS2101E\_S25\_A1. Make sure that all subfolders are checked: **.settings**, **bin**, and **src**. Under **To archive file**: browse to, e.g., desktop, and save it as **EECS2101E\_S25\_A1.zip**.

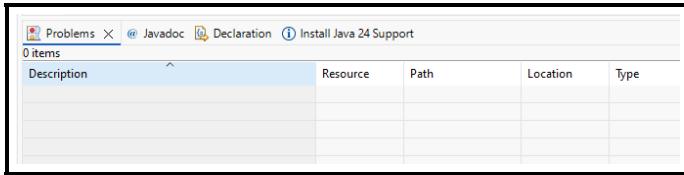


4. Then click **Finish**.

**Note.** In case you have concerns about exporting and submitting the **.setting** subfolder: it will be kept confidential and access-protected on eClass.

## 2 Submission

- Before you submit, you must make sure that the **Problems** panel on your Eclipse shows **no errors** (warnings are acceptable). In case you do not see the **Problems** panel: click on **Window**, then **Show View**, then **Problems**.



**Submitting programs with errors (meaning that it cannot be run for grading) will result in possible partial, but low, marks.**

- Section 1.3 asks you to **export** the Java project as an archive file:

**EECS2101E\_S25\_A1.zip**

Click on the following link (for which you will be prompted to enter your EECS account login credentials):

<https://webapp.eecs.yorku.ca/submit?acadyear=2024-25&term=S&course=2101E&assignment=a1>

- You **must** login into the web submit page using your EECS login credentials (otherwise, your submitted folder on the EECS server may not be identified properly):

The screenshot shows a login form titled "Web Submit Login". It instructs users to access Web Submit via their Passport York account or EECS account. It features fields for "EECS Username" and "EECS Password", both represented by empty input boxes. A "Login" button is at the bottom.

**Note.** If you are prompted for your PPY login instead, then it might be due to an earlier login session. In this case, login first with your PPY account credentials, then log out. Then, clicking on the above submission link should lead you to the login page for EECS account credentials.

- Ensure that the correct academic year (2024-25), term (S), course (2101E), and assignment (A1) are chosen. Then, browse to the archive file **EECS2101E\_S25\_A1.zip** and click on **Submit Files**.
- You may upload as many draft versions as you like before the deadline – only the latest submitted version of your work before the deadline will be graded.
- It is your **sole responsibility** to download and ensure that the submitted zip file is the one you intend to be graded (e.g., non-empty, not the starter project).

### **3 Amendments**

Clarifications or corrections will be added to this section.

-