# Assignment 4

Handout: **Tuesday, 1 November 2022**
Due: **11:59 pm, Friday, 11 November 2022**

Goals:

- To understand better the importance of information hiding, inheritance, polymorphism, and dynamic binding;
- To practice using monitor operations to coordinate the access to shared objects; and
- To get more familiar with the IntelliJ Idea IDE;

**Note**:

- In this assignment, we are specifying the timeout for the test in BankAccountTest.java using JUnit 5.7.0.

**Attention**

You MUST NOT change the names of the Java files or the signatures/return types of the methods in those files. We use tests to grade your assignments and the tests refer to the classes and methods defined in the given Java files, as is done in the released sample tests. Failed tests due to changes to the file names or method signatures/return types will be treated the same as failed tests due to incorrect implementations.

## 1. Generic Set (20 points)

Class `CompSet` implements a `HashSet` as described on page 18 of LEC07.pdf.

**What to Do:**

[Task 1] Read and complete the class so that all requirements expressed in both `CompSetTest.java` and the code comments are satisfied. Note that you are not allowed to add new fields to class `CompSet`.

## 2. RandomWalk (20 points)

This question deals with an undirected graph that consists of a set of nodes connected by edges. Each edge connects two different nodes, i.e., self-connection is not allowed. Given a set of random walk sequences obtained from this graph, you are required to recover some properties of this graph by referring to this set of random walk sequences. The definition of a random walk sequence has been given in the source code.

**What to Do:**

[Task 2] Read the classes `Graph.java` , `Node.java` and `RandomWalkSequence.java`. Complete the tasks in `Node.java` and `RandomWalkSequence.java` so that all requirements expressed in the code comments are satisfied.

**Hints:** To complete the tasks, you may need to use HashSet and HashMap, and the methods of these two classes.

## 3. BankAccount Using Monitor Operations (10 points)

In this task, you will need to use synchronized methods/statements and operations like `wait`, `notify`, and `notifyAll` to coordinate the `withdraw` and `deposite` operations on a shared bank account.

**What to Do:**

[Task 3] Complete class `BankAccount` so that the test in `BankAccountTest.java` always executes successfully.

**What to Hand in**

The whole Assignment4 folder with the completed methods in a ZIP file. The recommended procedures to obtain the ZIP file using IntelliJ IDEA: File -> Export -> Project to ZIP file.