

CSCI-UA 9102. Data structures

Summer session

Material for the Midterm

Augustin Cosse

June 5, 2026

1 Material covered (Theory)

1. You must be able to understand basic programming concepts including [Base types](#), [Strings](#), [Wrappers](#), [Arrays](#), [Enum Types](#)
2. You must be able to use and understand all the java [modifiers](#) including:
 - Access control modifiers: [public](#), [private](#), [protected](#)
 - The [static](#), [abstract](#) and [final](#) modifiers

You must be able to [write a class, or a method using those modifiers](#).

3. You must be able to understand and use [type conversion](#) (especially between Strings and numbers)
4. You must be able to know how to [import packages](#), [classes](#) and [methods from the classes](#) (e.g. the `Math` class or the `sqrt` method from the `Math` class)
5. You must be able to use and understand control flow in Java including [if else](#) and [switch](#) statements, [while](#) loops, [do while](#), [for](#) and [for each](#) loops as well as [break](#), [continue](#) and [return](#) statements.
6. You must be able to use and understand [how to prompt for user input](#) and [read command-line arguments](#) (i.e. `System.out.println`, `import java.util.Scanner` and `nextInt`, `nextDouble`,...)
7. You must be able to understand and explain [object oriented programming](#) (including the notion of constructor and the keyword `new`) and [inheritance](#) (including [abstract classes](#) and [interfaces](#), the keywords `extends` and `implements`)
8. You must be able to explain how to [catch](#) and [throw](#) an exception

9. You must be able to [use](#) and [manipulate multidimensional arrays](#).
10. You must be able to describe, compare and provide pseudo code for the implementation of [Singly Linked Lists](#), [Circularly Linked Lists](#) and [Doubly Linked Lists](#).

2 Material covered (Programming)

You must be able to solve all the exercises that have been solved in class. In particular:

1. For lab 1: Exercises 1.1a, 1.1b, 1.1c, 1.2a, 1.2b, 1.2d
2. For Lab 2: Exercises 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.8, 1.9, 1.12.
3. For Lab 3: Exercises 1.1, 1.3
4. For Lab 4: Exercises 1.1, 1.2, 3.1, 3.2, 3.3