In general tutorial sessions are to discuss the problems you faced during solving these exercises. Solutions will in general not be presented but discussed during these sessions.

**Exercise 1  Gradiance – Compulsory! Hurdle!**

Sign up to the course (if you haven’t already) on Gradiance with Class Token: **8DDCA614**. Please log in and complete the assigned homework “Chapter 08: Turing Machines”.

**Exercise 2  Turing Machine Design**

Design Turing Machines for the following languages. (Using JFLAP is recommended.)

1. The set of strings with an equal number of 0’s and 1’s.
2. \( \{a^n b^n c^n \mid n \geq 1\} \)

**Exercise 3  Turing-completeness**

Compile a list of computation models that you know to be Turing-complete (able to simulate a deterministic single-tape Turing machines).

**Exercise 4  TMs for partial functions**

A function is called partial if it may be undefined for some arguments. A Turing machine \( M \) is said to compute a partial function \( f : \mathbb{N} \to \mathbb{N} \) if \( M \) halts on all inputs \( x \) for which \( f \) is defined and outputs \( f(x) \).

(a) Show how, given a TM that computes \( f \), you can construct a TM that accepts the graph of \( f \) as a language.

(b) Show how, given a TM that accepts the graph of \( f \), you can construct a TM that computes \( f \).