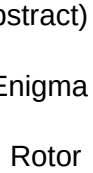
Enigma Emulator - Task 1 Object-Oriented Design

1. Overview
This document describes an object-oriented design for an Enigma M3 emulator. The design follows the hierarchy suggested in the assignment: a common abstract Substitutor class, a Translator class for simple permutations, and concrete components Reflector, Plugboard, Rotor, and Enigma.

2. Class Hierarchy (UML-style diagram)



Arrows denote inheritance. Enigma has (composition) three Rotors, one Reflector, and one Plugboard as internal fields.

3. Class Responsibilities

Substitutor:
- Abstract base class for all components that perform letter substitution.
- Provides helper methods for letter-index conversion (A-Z <-> 0-25).
- Provides circular shift helper: (index + delta) mod 26.
- Declares abstract methods translate_forward(c) and translate_reverse(c).

Translator (extends Substitutor):
- Represents a simple permutation over the alphabet.
- Holds a forward permutation of size 26 (array or string).
- Computes the reverse permutation automatically from the forward one.
- Implements translate_forward(c) using the forward permutation.
- Implements translate_reverse(c) using the reverse permutation.

Reflector (extends Translator):
- Models the Enigma reflector.
- Uses a symmetric permutation: if A -> Y then Y -> A.
- Because of symmetry, forward and reverse mappings are identical.

Plugboard (extends Translator):
- Models the plugboard at the front of the machine.
- Is configured from up to 10 letter pairs (e.g., 'AT CE RL').
- Builds a full 26-letter permutation where unpaired letters map to themselves.

Rotor (extends Translator):
- Models a single rotating Enigma rotor.
- Holds the internal wiring (forward permutation) and its computed reverse.
- Additional state: ring_setting (0-25), offset (0-25), and notch position.
- Method step(): advances the offset by one (offset = (offset + 1) mod 26).
- Method is_at_notch(): true when current window letter equals the notch letter.
- translate_forward(c): applies ring_setting and offset before and after wiring.
- translate_reverse(c): same idea but uses the reverse permutation.

Enigma (extends Substitutor):
- Represents the complete Enigma machine.
- Has three Rotor objects (left, middle, right), one Reflector, and one Plugboard.
- configure(...): chooses rotor types, ring settings, initial offsets, and plugboard.
- step_rotors(): implements the single- and double-stepping behaviour: