

Lab4

Submission Details, Grading Criteria, and Demo/Viva Policy

Submission Details

Students must submit a **single compressed archive** named:

```
lab4_<rollno>.tar.gz
```

Expected Contents

The submission should include:

- 1. **VM Implementation**
 - Stack-based bytecode virtual machine supporting the defined instruction set (as per Lab 4 spec).
 - Correct handling of:
 - stack operations
 - arithmetic/logical instructions
 - control flow
 - memory access
 - function calls and returns
 - Deterministic execution and no memory leaks.
- 2. **Assembler / Bytecode Generator**
 - Tool to convert human-readable assembly into bytecode format.
 - Clear instruction encoding consistent with the VM.
- 3. **Test Programs**
 - Multiple bytecode programs demonstrating:
 - arithmetic expressions
 - loops and branching
 - function calls
 - memory load/store
 - At least one non-trivial program (e.g., iterative computation, nested calls).
- 4. **Benchmarks / Performance Notes**
 - Basic benchmarks or timing notes for representative programs.
 - Comparison across instruction counts or program sizes is sufficient.
- 5. **Technical Report**
 - Architecture of the VM (stack, PC, memory model).
 - Instruction dispatch strategy.
 - Design of call frames and return mechanism.
 - Discussion of limitations and possible enhancements.
- 6. **README**
 - Build and execution instructions.
 - How to run assembler and VM.
 - How to execute provided test programs.

Grading Criteria (15 Marks Total)

A. Core VM Functionality (7 marks)

- Correct implementation of instruction set: stack ops, arithmetic, control flow (3)
- Proper call/return handling and stack frame management (2)
- Bytecode loading and execution correctness (2)

B. Assembler / Tooling (3 marks)

- Correct translation from assembly to bytecode (2)
- Robust handling of invalid instructions / operands (1)

C. System Design & Code Quality (3 marks)

- Clean modular design (VM core, instruction handling, memory, assembler) (1.5)
- Extensibility and clarity of implementation (1.5)

D. Documentation & Tests (2 marks)

- Quality of report and explanation (1)
- Coverage and correctness of test programs (1)

Optional enhancements (e.g., JIT compilation, standard library) may be considered positively but marks are capped at 15.

Demo / Viva Evaluation

- Demos will be conducted **in person by the TAs**.
 - During the demo, students will be asked to:
 - Run their VM on provided bytecode programs
 - Execute additional test cases supplied by the TA
 - Explain VM architecture, instruction handling, and control flow
 - Students must be able to explain their code clearly; use of AI tools is allowed, but lack of understanding will result in deductions.
-

Attendance Policy for Demo/Viva

- Attendance will be recorded during the demo/viva.
- For **pairs**:
 - If **both partners attend**: both are evaluated normally.
 - If **one partner is absent**:
 - Absent student: **50% of assignment marks, 0 marks in viva**
 - Present partner: **100% evaluation for both assignment and viva**
- Attendance status (present/absent) will be explicitly recorded in the grading sheet.