

Lab 5 – Mark-Sweep Garbage Collector

Submission Instructions

Students must submit a **single compressed archive** named: lab5_rolno.tar.gz

Expected Contents

The submission must include:

1. GC-enabled VM

- The VM from Lab 4 extended with a **stop-the-world mark-sweep garbage collector**.
- Support for:
 - heap allocation
 - root discovery (VM stack as root set)
 - mark phase
 - sweep phase
- GC must be callable explicitly (e.g., `gc(vm)`).

2. Source Code

- Clear integration of GC with the VM.
- No memory leaks or unsafe memory access.
- Code must compile and run on the lab environment.

3. Test Programs

- Programs covering:
 - reachable vs unreachable objects
 - transitive reachability
 - cyclic references
 - deep object graphs
 - stress allocation
- Provided test cases may be used, but additional tests are encouraged.

4. Performance / Stress Evaluation

- Brief evaluation showing GC behavior under heavy allocation.
- Simple metrics (execution time, number of objects freed, etc.) are sufficient.

5. Technical Report

- Description of GC design and integration.
- Root identification strategy.
- Mark and sweep implementation details.
- Discussion of correctness and limitations.

6. README

- Build instructions.
 - How to run the VM and trigger GC.
 - How to execute test cases.
-

Demo and Evaluation

- Demos will be conducted by the TAs.
- Students will be asked to:
 - Run GC on provided and TA-supplied test cases
 - Explain GC design, root handling, and object traversal
- Code may be tested under **stress and edge cases** not included in the submission.

Failure to explain the implementation may result in loss of marks.