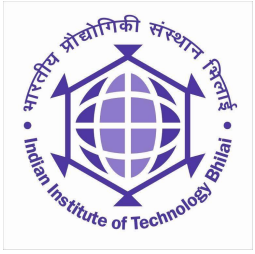


DEPARTMENT OF EECS

Indian Institute of Technology Bhilai

CS200 — SOFTWARE TOOLS AND TECHNOLOGIES Lab II

Scope: Git Object Model

Difficulty Level: Beginner

Assignment 1
September 12, 2020

- Instructions
  - All answers will be in separate files in a single folder, named: `<group-id>_<group-name>`
  - Name files as `q<question-no>` without any extension. e.g., q2
  - Use LaTeX to show your answers that need `git` graphs
  - Make a `tarball` for the folder that contains your answers
  - Compress the `tarball` using `gzip` before uploading on Piazza

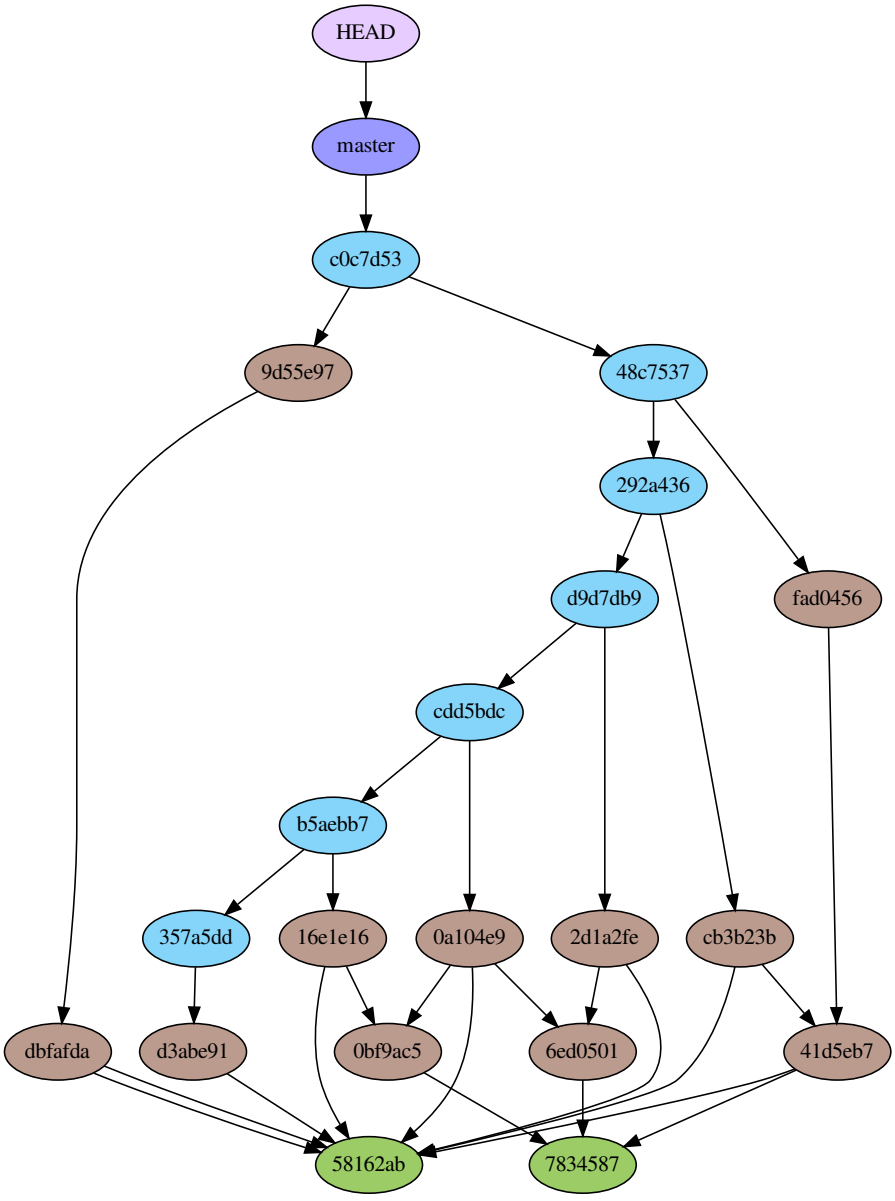
1. Write a shell script for the following [Warm-up]
   (a) Initialize `git`.
   (b) Create a file with content as *iitbhilai* name it as `<roll-no-member-1>`.
   (c) Now copy the same file w.r.t to the number of members of your group and rename as `<roll-no-member-2>`, `<roll-no-member-3>`.
   (d) Put all three files into three individual folders with your names `<name-1>`, `<name-2>`, `<name-3>` on them.
   (e) Commit all the files **at once**.
   (f) Show the `git-graph`.
2. Redo Problem 1 but with incremental commits. You have to show the `git-graph` after every commit. Submit a LaTeX generated `pdf` which will depict how the `git` repository evolved.
3. Write a shell script for the following [Git-Objects]
   (a) For all files in `.git/objects` directory in Problem 2 run a loop to show the type of each object. (Find the git command that shows the type of a `git-object`)
   (b) If the object is a `blob`, show the content.
   (c) If the object is a `tree`, show all blobs it points to.
   (d) If the object is a `commit`, then show the author and parent.

   You are allowed to use `git` native commands to extract information.

   [Hint: You need to understand how the directory is managed by `git`.]
4. List the sequence of `shell/git` commands that generates the following `git` graph. [Git-graph]

   Incrementally show the `git` graph after every commit listing the `shell/git` commands executed before it.

Question 4 continues on the next page...



Note: You do not need to produce the *same* hash as shown, but recreate the structure.
