Commit requirements

Every commit in the work process should follow the code style and processual rules. This is mandatory for the fast and smooth workflow.

To create a commit you need to:

```
< make changes >
$ git diff
$ git add < files >
$ git status
$ git commit
$ git show
```

git status command needed to check the current branch for commit and the current state of the repository.
git diff and git show commands and needed to ensure that changes are valid.

- Use signature when committing, this will be automatically done by the
  $ git commit -s (name needs to be set in ~/.gitconfig).
- No more than 72 characters on line (use your signature string as a basic ruler).
- Empty line between title and description and between description and signature.

Commit title should be like:

Topic: <Action> <the rest title text>
- Topic: describes the area of changes, e.g. Lesson1: or LCD:
- Action: Add, Fix, Implement … (not added, implemented). Should start from capital letter.
- The whole title should give enough info about the commit when using $ git log --oneline.
- No dot at the end of the title.Commit description should describe why and how it was done, and not what was done (this is visible in diff).

For example, a bad description: Changed short to int.

The correct one should be: Moved to a bigger data type because of overflow in some corner cases.

Description is not needed in the “title says it all” cases.

- Do not mix several different actions in one commit: split them in several. For example, code style fixes and the change itself.
- Branch at each commit should be buildable and working. Do not add constants (header) and their usage (code) in different commits.