

Before:

```
1  #pragma once
2  #include <iostream>
3
4  class SingleClass; //^^las^^ class SingleHunkClass;
5
6  struct SubstitutionStruct{}; //^^las^^
7  //^^las sub begin^^
8
9  double value{};
10  √ ////this is comment example
11  //^^las sub end^^
12
13
14  √ int getMax(int a, int b)
15  {
16  √   if (a > b)
17  |   {
18  |       return a;
19  |   }
20  |   return b;
21  }
22
23
24  //^^las begin^^
25  √ int fun()
26  {
27  |   double c = 0.2;
28  |   return c;
29  }
30
31  √ //^^las end^^
32  //^^las sub begin^^
33  √ void substFun()
34  {
35  |   std::cout<<"Function comes from substitution LasHunk"<<std::endl;
36  |   }
37  //^^las sub end^^
```

After:

```
1  #pragma once
2  #include <iostream>
3
4  class SingleHunkClass;
5
6
7  double value{};
8  //this is comment example
9
10
11  int getMax(int a, int b)
12  {
13      if (a > b)
14      {
15          return a;
16      }
17      return b;
18  }
19
20
21  void substFun()
22  {
23      std::cout<<"Function comes from substitution LasHunk"<<std::endl;
24  }
```

"Las (Log, Analyze and Share) app

The Las tool is useful for removing temporary code and substitute it with final version when developing new features, debugging issues etc.

If you want to commit changes, you don't need to check and clear every file to remove unwanted code. You can share with others clean code version using diff files without actually changing your current code.

Las tags can be used with multiple programming languages or with plain text files. Available tags should be prefixed with given language comment or # for plain text:

Removes code between `^^las begin^^` and `^^las end^^` or single line tagged with `^^las^^`.

Optionally substitutes removed code with changes enclosed by `^^las sub begin^^` and `^^las sub end^^` or for single line by the code following `^^las^^` in that line.

Usage for CPP language, where // is comment(for other langs substitute \"//\" for corresponding comment token):

```
//^\\las begin^^  
code  
//^\\las end^^  
//^\\las sub begin^^  
code - often commented  
//^\\las sub end^^
```

or
line//^\\las^^ substitution code

or
line//^\\las^^
//^\\las sub begin^^
code - often commented
//^\\las sub end^^

Las commands:\\n

- las cut [filenames]
e.g. las cut ; las cut file1 file2
Remove code/add substitution code marked with LAS tags.
Command changes files content. \\n
- las diff [filenames]
e.g. las diff ; las diff file1 file2
Create diff output from removed code/added substitution code marked with LAS tags
in files from working area.
Command keeps files content untouched.\\n
- las diff head [filenames]
e.g. las diff head ; las diff head file1 file2
Create diff output from removed code/added substitution code marked with LAS tags
in files from working and staging area.
Command keeps files content untouched.\\n
- las diff staged [filenames]
e.g. las diff staged ; las diff staged file1 file2
Create diff output from removed code/added substitution code marked with LAS tags
in files from staging area.
Command keeps files content untouched.\\n
- las restore -n<number>
e.g. las restore -n0 \\n
Command allows to restore state of repository as before executing given las command.
<number> is related to given snapshot and can be obtained using \"las restore --list\".
Command changes files content and must be run on repository without uncommitted changes.\\n
- las restore --list
Command shows list of already taken repository's snapshots.
Command keeps files content untouched.\\n
- las restore --show -n<number>
e.g. las restore --show -n2
Command show changes related to given snapshot, where <number> can be obtained using \"las restore --list\".
Command keeps files content untouched.\\n "