u115-05@PROG-05 MINGW32 ~/Documents/lab02

$ mkdir bob

u115-05@PROG-05 MINGW32 ~/Documents/lab02

$ cd alice/project/

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project

$ git init

Initialized empty Git repository in C:/Users/u115-05/Documents/lab02/alice/proje

ct/.git/

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git config user.name 'Alice (TrofimenkoAI)'

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git config user.email 'alice@example.com'

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add ..." to include in what will be committed)

bin/

main.cpp

obj/

project.cbp

nothing added to commit but untracked files present (use "git add" to track)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached ..." to unstage)

new file: main.cpp

Untracked files:

(use "git add ..." to include in what will be committed)

bin/

obj/

project.cbp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m 'code'

[master (root-commit) 8ca0283] code

1 file changed, 9 insertions(+)

create mode 100644 main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ ^C

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add project.cbp

warning: LF will be replaced by CRLF in project.cbp.

The file will have its original line endings in your working directory

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m 'build: add project`s file '

[master 92b82ee] build: add project`s file

1 file changed, 43 insertions(+)

create mode 100644 project.cbp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m 'zamena tela main.cpp '

On branch master

Untracked files:

bin/

obj/

nothing added to commit but untracked files present

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add -u

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m 'summ & raz'

On branch master

Changes not staged for commit:

modified: main.cpp

Untracked files:

bin/

obj/

no changes added to commit

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -a -m "summ & raz"

[master bb12baf] summ & raz

1 file changed, 4 insertions(+), 1 deletion(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

(use "git add ..." to include in what will be committed)

.gitignore

Untitled1.c

obj/

project.depend

nothing added to commit but untracked files present (use "git add" to track)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

(use "git add ..." to include in what will be committed)

.gitignore

obj/

project.depend

nothing added to commit but untracked files present (use "git add" to track)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

(use "git add ..." to include in what will be committed)

.gitignore

obj/

project.depend

nothing added to commit but untracked files present (use "git add" to track)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

(use "git add ..." to include in what will be committed)

.gitignore

project.depend

nothing added to commit but untracked files present (use "git add" to track)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add -u

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add .gitigonre

fatal: pathspec '.gitigonre' did not match any files

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add .gitignore

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit "git: ignore "

error: pathspec 'git: ignore ' did not match any file(s) known to git

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit 'git: ignoring'

error: pathspec 'git: ignoring' did not match any file(s) known to git

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m "git: ignoring"

[master cdc23b2] git: ignoring

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log --main.cpp

fatal: unrecognized argument: --main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log -- main.cpp

commit bb12bafc73f3d2d81e7bf5875403d8fc678b02cd

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:28:09 2020 +0300

summ & raz

commit 8ca0283786cce22eb2ee7deab9f895c6ba9e49fd

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:01:50 2020 +0300

code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log --grep "code"

commit 8ca0283786cce22eb2ee7deab9f895c6ba9e49fd

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:01:50 2020 +0300

code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log --oneline

cdc23b2 (HEAD -> master) git: ignoring

bb12baf summ & raz

92b82ee build: add project`s file

8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log--oneline --decorate --all --graph

git: 'log--oneline' is not a git command. See 'git --help'.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log --oneline --decorate --all --graph

\* cdc23b2 (HEAD -> master) git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git show HEAD

commit cdc23b2b6ae67d77dfa402a09e37011a67004e9d (HEAD -> master)

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:51:18 2020 +0300

git: ignoring

diff --git a/.gitignore b/.gitignore

new file mode 100644

index 0000000..4c7473d

--- /dev/null

+++ b/.gitignore

@@ -0,0 +1,2 @@

+/bin

+/obj

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git show HEAD~1

commit bb12bafc73f3d2d81e7bf5875403d8fc678b02cd

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:28:09 2020 +0300

summ & raz

diff --git a/main.cpp b/main.cpp

index b4392ec..44bb5b1 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,9 @@ using namespace std;

int main()

{

- cout << "Hello world!" << endl;

+ int a, b;

+ cout << "Enter A, B" << endl;

+ cin >> a >> b;

+ cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git show master~1

commit bb12bafc73f3d2d81e7bf5875403d8fc678b02cd

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:28:09 2020 +0300

summ & raz

diff --git a/main.cpp b/main.cpp

index b4392ec..44bb5b1 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,9 @@ using namespace std;

int main()

{

- cout << "Hello world!" << endl;

+ int a, b;

+ cout << "Enter A, B" << endl;

+ cin >> a >> b;

+ cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ gi show bb12baf

bash: gi: command not found

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git show bb12baf

commit bb12bafc73f3d2d81e7bf5875403d8fc678b02cd

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:28:09 2020 +0300

summ & raz

diff --git a/main.cpp b/main.cpp

index b4392ec..44bb5b1 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,9 @@ using namespace std;

int main()

{

- cout << "Hello world!" << endl;

+ int a, b;

+ cout << "Enter A, B" << endl;

+ cin >> a >> b;

+ cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git diff

diff --git a/main.cpp b/main.cpp

index 44bb5b1..8b950e9 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,5 +8,7 @@ int main()

cout << "Enter A, B" << endl;

cin >> a >> b;

cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

+ cout << "A \* B = " << a \* b << '\n';

+

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git diff HEAD~2

diff --git a/.gitignore b/.gitignore

new file mode 100644

index 0000000..4c7473d

--- /dev/null

+++ b/.gitignore

@@ -0,0 +1,2 @@

+/bin

+/obj

diff --git a/main.cpp b/main.cpp

index b4392ec..8b950e9 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,11 @@ using namespace std;

int main()

{

- cout << "Hello world!" << endl;

+ int a, b;

+ cout << "Enter A, B" << endl;

+ cin >> a >> b;

+ cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

+ cout << "A \* B = " << a \* b << '\n';

+

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git diff HEAD~2 HEAD

diff --git a/.gitignore b/.gitignore

new file mode 100644

index 0000000..4c7473d

--- /dev/null

+++ b/.gitignore

@@ -0,0 +1,2 @@

+/bin

+/obj

diff --git a/main.cpp b/main.cpp

index b4392ec..44bb5b1 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,9 @@ using namespace std;

int main()

{

- cout << "Hello world!" << endl;

+ int a, b;

+ cout << "Enter A, B" << endl;

+ cin >> a >> b;

+ cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m "code: proizvedenie"

[master c98877d] code: proizvedenie

1 file changed, 2 insertions(+)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ get reset --hard HEAD~1

bash: get: command not found

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git reset --hard HEAD~1

HEAD is now at cdc23b2 git: ignoring

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git show HEAD

commit cdc23b2b6ae67d77dfa402a09e37011a67004e9d (HEAD -> master)

Author: Alice (TrofimenkoAI)

Date: Fri Mar 13 10:51:18 2020 +0300

git: ignoring

diff --git a/.gitignore b/.gitignore

new file mode 100644

index 0000000..4c7473d

--- /dev/null

+++ b/.gitignore

@@ -0,0 +1,2 @@

+/bin

+/obj

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git checkout HEAD -- main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git remote add origin https://github.com/TrofimenkoAI/cs-lab02.git

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git push -u origin master

TrEnumerating objects: 12, done.

Counting objects: 100% (12/12), done.

Compressing objects: 100% (10/10), done.

Writing objects: 100% (12/12), 1.62 KiB | 207.00 KiB/s, done.

Total 12 (delta 0), reused 0 (delta 0)

To https://github.com/TrofimenkoAI/cs-lab02.git

\* [new branch] master -> master

Branch 'master' set up to track remote branch 'master' from 'origin'.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git commit -m "code: max"

[master 48dd832] code: max

1 file changed, 14 insertions(+), 4 deletions(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git diff

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 502 bytes | 251.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/TrofimenkoAI/cs-lab02.git

08d5112..48dd832 master -> master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git fitch

git: 'fitch' is not a git command. See 'git --help'.

The most similar command is

fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/TrofimenkoAI/cs-lab02

48dd832..71c88c6 master -> origin/master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git pull --f-only

error: unknown option `f-only'

usage: git pull [] [ [...]]

-v, --verbose be more verbose

-q, --quiet be more quiet

--progress force progress reporting

--recurse-submodules[=]

control for recursive fetching of submodules

Options related to merging

-r, --rebase[=(false|true|merges|preserve|interactive)]

incorporate changes by rebasing rather than merging

-n do not show a diffstat at the end of the merge

--stat show a diffstat at the end of the merge

--log[=] add (at most ) entries from shortlog to merge commi

t message

--signoff[=...] add Signed-off-by:

--squash create a single commit instead of doing a merge

--commit perform a commit if the merge succeeds (default)

--edit edit message before committing

--ff allow fast-forward

--ff-only abort if fast-forward is not possible

--verify-signatures verify that the named commit has a valid GPG signature

--autostash automatically stash/stash pop before and after rebase

-s, --strategy

merge strategy to use

-X, --strategy-option

option for selected merge strategy

-S, --gpg-sign[=]

GPG sign commit

--allow-unrelated-histories

allow merging unrelated histories

Options related to fetching

--all fetch from all remotes

-a, --append append to .git/FETCH\_HEAD instead of overwriting

--upload-pack path to upload pack on remote end

-f, --force force overwrite of local branch

-t, --tags fetch all tags and associated objects

-p, --prune prune remote-tracking branches no longer on remote

-j, --jobs[=] number of submodules pulled in parallel

--dry-run dry run

-k, --keep keep downloaded pack

--depth deepen history of shallow clone

--unshallow convert to a complete repository

--update-shallow accept refs that update .git/shallow

--refmap specify fetch refmap

-4, --ipv4 use IPv4 addresses only

-6, --ipv6 use IPv6 addresses only

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git getch

git: 'getch' is not a git command. See 'git --help'.

The most similar command is

fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$

Display all 3027 possibilities? (y or n)

!

./

:

[

[.exe

[[

]]

\_\_git

\_\_git\_aliased\_command

\_\_git\_complete

\_\_git\_complete\_command

\_\_git\_complete\_common

\_\_git\_complete\_fetch\_refspecs

\_\_git\_complete\_file

\_\_git\_complete\_force\_with\_lease

\_\_git\_complete\_index\_file

\_\_git\_complete\_refs

\_\_git\_complete\_remote\_or\_refspec

\_\_git\_complete\_revlist

\_\_git\_complete\_revlist\_file

\_\_git\_complete\_strategy

\_\_git\_complete\_symbol

\_\_git\_compute\_all\_commands

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$

Display all 3027 possibilities? (y or n)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$

Display all 3027 possibilities? (y or n)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log --oneline --decorate --all --graph

\* 71c88c6 (origin/master) code: max & min

\* 48dd832 (HEAD -> master) code: max

\* 08d5112 code: delenie

\* 2ff2261 code: proizvedenie

\* cdc23b2 git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git pull -ff-only

error: did you mean `--ff-only` (with two dashes ?)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git pull --ff-only

Updating 48dd832..71c88c6

Fast-forward

main.cpp | 17 +++++++++++++----

1 file changed, 13 insertions(+), 4 deletions(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git branch double

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git checkout double

Switched to branch 'double'

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (double)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (double)

$ git commit -m "code: double"

[double 2c6ea50] code: double

1 file changed, 1 insertion(+), 1 deletion(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (double)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$

Display all 3027 possibilities? (y or n)

!

./

:

[

[.exe

[[

]]

\_\_git

\_\_git\_aliased\_command

\_\_git\_complete

\_\_git\_complete\_command

\_\_git\_complete\_common

\_\_git\_complete\_fetch\_refspecs

\_\_git\_complete\_file

\_\_git\_complete\_force\_with\_lease

\_\_git\_complete\_index\_file

\_\_git\_complete\_refs

\_\_git\_complete\_remote\_or\_refspec

\_\_git\_complete\_revlist

\_\_git\_complete\_revlist\_file

\_\_git\_complete\_strategy

\_\_git\_complete\_symbol

\_\_git\_compute\_all\_commands

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git merge double

Updating 71c88c6..2c6ea50

Fast-forward

main.cpp | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git log --oneline --decorate --all --graph

\* 2c6ea50 (HEAD -> master, double) code: double

\* 71c88c6 (origin/master) code: max & min

\* 48dd832 code: max

\* 08d5112 code: delenie

\* 2ff2261 code: proizvedenie

\* cdc23b2 git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 358 bytes | 179.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/TrofimenkoAI/cs-lab02.git

71c88c6..2c6ea50 master -> master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/alice/project (master)

$

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob

$ https://github.com/TrofimenkoAI/cs-lab02.git

bash: https://github.com/TrofimenkoAI/cs-lab02.git: No such file or directory

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob

$ git clone https://github.com/TrofimenkoAI/cs-lab02.git project

Cloning into 'project'...

remote: Enumerating objects: 12, done.

remote: Counting objects: 100% (12/12), done.

remote: Compressing objects: 100% (10/10), done.

remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0

Unpacking objects: 100% (12/12), done.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob

$ cd project

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git config

usage: git config []

Config file location

--global use global config file

--system use system config file

--local use repository config file

--worktree use per-worktree config file

-f, --file use given config file

--blob read config from given blob object

Action

--get get value: name [value-regex]

--get-all get all values: key [value-regex]

--get-regexp get values for regexp: name-regex [value-regex]

--get-urlmatch get value specific for the URL: section[.var] URL

--replace-all replace all matching variables: name value [value\_rege

x]

--add add a new variable: name value

--unset remove a variable: name [value-regex]

--unset-all remove all matches: name [value-regex]

--rename-section rename section: old-name new-name

--remove-section remove a section: name

-l, --list list all

-e, --edit open an editor

--get-color find the color configured: slot [default]

--get-colorbool find the color setting: slot [stdout-is-tty]

Type

-t, --type <> value is given this type

--bool value is "true" or "false"

--int value is decimal number

--bool-or-int value is --bool or --int

--path value is a path (file or directory name)

--expiry-date value is an expiry date

Other

-z, --null terminate values with NUL byte

--name-only show variable names only

--includes respect include directives on lookup

--show-origin show origin of config (file, standard input, blob, com

mand line)

--default with --get, use default value when missing entry

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git commit -m "code: proizvedenie"

[master 2ff2261] code: proizvedenie

Committer: u115-05

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 2 insertions(+)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ gi show head

bash: gi: command not found

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git show HEAD

commit 2ff2261fac4f73f1d963c0dbae6c28452e458b5a (HEAD -> master)

Author: u115-05

Date: Fri Mar 13 11:36:33 2020 +0300

code: proizvedenie

diff --git a/main.cpp b/main.cpp

index 44bb5b1..8b950e9 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,5 +8,7 @@ int main()

cout << "Enter A, B" << endl;

cin >> a >> b;

cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

+ cout << "A \* B = " << a \* b << '\n';

+

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 387 bytes | 193.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/TrofimenkoAI/cs-lab02.git

cdc23b2..2ff2261 master -> master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/TrofimenkoAI/cs-lab02

2ff2261..08d5112 master -> origin/master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git log --oneline --decorate --all --graph

\* 08d5112 (origin/master, origin/HEAD) code: delenie

\* 2ff2261 (HEAD -> master) code: proizvedenie

\* cdc23b2 git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ get pull --ff-only

bash: get: command not found

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull --ff-only

Updating 2ff2261..08d5112

Fast-forward

main.cpp | 1 +

1 file changed, 1 insertion(+)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git commit -m "code: norm vid"

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git commit -m "code: norm vid"

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git push

Everything up-to-date

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git commit -m "code: min"

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull--ff-only

git: 'pull--ff-only' is not a git command. See 'git --help'.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull --ff-only

Already up to date.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull --ff-only

Already up to date.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git commit -m "code: min"

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git diff

diff --git a/main.cpp b/main.cpp

index f1e481e..ef4c34e 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,12 +4,22 @@ using namespace std;

int main()

{

diff --git a/main.cpp b/main.cpp

index f1e481e..ef4c34e 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,12 +4,22 @@ using namespace std;

int main()

{

int main()

{

diff --git a/main.cpp b/main.cpp

index f1e481e..ef4c34e 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,12 +4,22 @@ using namespace std;

int main()

{

- int a, b;

+ int a, b, min;

cout << "Enter A, B" << endl;

cin >> a >> b;

- cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

- cout << "A \* B = " << a \* b << '\n';

- cout << "A / B = " << a / b << '\n';

+

+

+ if (a < b)

+ min = a;

+ else

+ min = b;

+

+

+ cout << "A + B = " << a + b << '\n'

+ << "A - B = " << a - b << '\n'

+ << "A \* B = " << a \* b << '\n'

+ << "A / B = " << a / b << '\n';

+

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git commit -m "code: min"

[master 3fe379d] code: min

Committer: u115-05

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 14 insertions(+), 4 deletions(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git push

To https://github.com/TrofimenkoAI/cs-lab02.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/TrofimenkoAI/cs-lab02.git

'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/TrofimenkoAI/cs-lab02

08d5112..48dd832 master -> origin/master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull --ff-only

fatal: Not possible to fast-forward, aborting.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git show master

commit 3fe379dd2dbfa1b46d23c018fb6df503a0da9f35 (HEAD -> master)

Author: u115-05

Date: Fri Mar 13 12:08:16 2020 +0300

code: min

diff --git a/main.cpp b/main.cpp

index f1e481e..ef4c34e 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,12 +4,22 @@ using namespace std;

int main()

{

- int a, b;

+ int a, b, min;

cout << "Enter A, B" << endl;

cin >> a >> b;

- cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

- cout << "A \* B = " << a \* b << '\n';

- cout << "A / B = " << a / b << '\n';

+

+

q

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull --ff-only

fatal: Not possible to fast-forward, aborting.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git fench

git: 'fench' is not a git command. See 'git --help'.

The most similar command is

fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git fetch

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git pull --ff-only

fatal: Not possible to fast-forward, aborting.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git rebase origin/master

First, rewinding head to replay your work on top of it...

Applying: code: min

Using index info to reconstruct a base tree...

M main.cpp

Falling back to patching base and 3-way merge...

Auto-merging main.cpp

CONFLICT (content): Merge conflict in main.cpp

error: Failed to merge in the changes.

hint: Use 'git am --show-current-patch' to see the failed patch

Patch failed at 0001 code: min

Resolve all conflicts manually, mark them as resolved with

"git add/rm ", then run "git rebase --continue".

You can instead skip this commit: run "git rebase --skip".

To abort and get back to the state before "git rebase", run "git rebase --abort"

.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git show master

commit 3fe379dd2dbfa1b46d23c018fb6df503a0da9f35 (master)

Author: u115-05

Date: Fri Mar 13 12:08:16 2020 +0300

code: min

diff --git a/main.cpp b/main.cpp

index f1e481e..ef4c34e 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,12 +4,22 @@ using namespace std;

int main()

{

- int a, b;

+ int a, b, min;

cout << "Enter A, B" << endl;

cin >> a >> b;

- cout << "A + B = " << a + b << '\n' << "A - B = " << a - b << '\n';

- cout << "A \* B = " << a \* b << '\n';

- cout << "A / B = " << a / b << '\n';

+

+

+ if (a < b)

+ min = a;

+ else

+ min = b;

+

+

+ cout << "A + B = " << a + b << '\n'

+ << "A - B = " << a - b << '\n'

+ << "A \* B = " << a \* b << '\n'

+ << "A / B = " << a / b << '\n';

+

return 0;

}

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ gie commit -m "code: max & min"

bash: gie: command not found

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$

Display all 3027 possibilities? (y or n)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git commit -m "code: max & min"

[detached HEAD 71c88c6] code: max & min

Committer: u115-05

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly. Run the

following command and follow the instructions in your editor to edit

your configuration file:

git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 13 insertions(+), 4 deletions(-)

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git rebase --continue

Applying: code: min

No changes - did you forget to use 'git add'?

If there is nothing left to stage, chances are that something else

already introduced the same changes; you might want to skip this patch.

Resolve all conflicts manually, mark them as resolved with

"git add/rm ", then run "git rebase --continue".

You can instead skip this commit: run "git rebase --skip".

To abort and get back to the state before "git rebase", run "git rebase --abort"

.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git log --oneline --decorate --all --graph

\* 71c88c6 (HEAD) code: max & min

\* 48dd832 (origin/master, origin/HEAD) code: max

| \* 3fe379d (master) code: min

|/

\* 08d5112 code: delenie

\* 2ff2261 code: proizvedenie

\* cdc23b2 git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git rebase origin/master

fatal: It seems that there is already a rebase-apply directory, and

I wonder if you are in the middle of another rebase. If that is the

case, please try

git rebase (--continue | --abort | --skip)

If that is not the case, please

rm -fr ".git/rebase-apply"

and run me again. I am stopping in case you still have something

valuable there.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$

Display all 3027 possibilities? (y or n)

!

./

:

[

[.exe

[[

]]

\_\_git

\_\_git\_aliased\_command

\_\_git\_complete

\_\_git\_complete\_command

\_\_git\_complete\_common

\_\_git\_complete\_fetch\_refspecs

\_\_git\_complete\_file

\_\_git\_complete\_force\_with\_lease

\_\_git\_complete\_index\_file

\_\_git\_complete\_refs

\_\_git\_complete\_remote\_or\_refspec

\_\_git\_complete\_revlist

\_\_git\_complete\_revlist\_file

\_\_git\_complete\_strategy

\_\_git\_complete\_symbol

\_\_git\_compute\_all\_commands

\_\_git\_compute\_config\_vars

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git add main.cpp

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git rebase --continue

Applying: code: min

No changes - did you forget to use 'git add'?

If there is nothing left to stage, chances are that something else

already introduced the same changes; you might want to skip this patch.

Resolve all conflicts manually, mark them as resolved with

"git add/rm ", then run "git rebase --continue".

You can instead skip this commit: run "git rebase --skip".

To abort and get back to the state before "git rebase", run "git rebase --abort"

.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git log --oneline --decorate --all --graph

\* 71c88c6 (HEAD) code: max & min

\* 48dd832 (origin/master, origin/HEAD) code: max

| \* 3fe379d (master) code: min

|/

\* 08d5112 code: delenie

\* 2ff2261 code: proizvedenie

\* cdc23b2 git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master|REBASE 1/1)

$ git rebase --skip

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git rebase origin/master

Current branch master is up to date.

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git log --oneline --decorate --all --graph

\* 71c88c6 (HEAD -> master) code: max & min

\* 48dd832 (origin/master, origin/HEAD) code: max

\* 08d5112 code: delenie

\* 2ff2261 code: proizvedenie

\* cdc23b2 git: ignoring

\* bb12baf summ & raz

\* 92b82ee build: add project`s file

\* 8ca0283 code

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 422 bytes | 211.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/TrofimenkoAI/cs-lab02.git

48dd832..71c88c6 master -> master

u115-05@PROG-05 MINGW32 ~/Documents/lab02/bob/project (master)

$