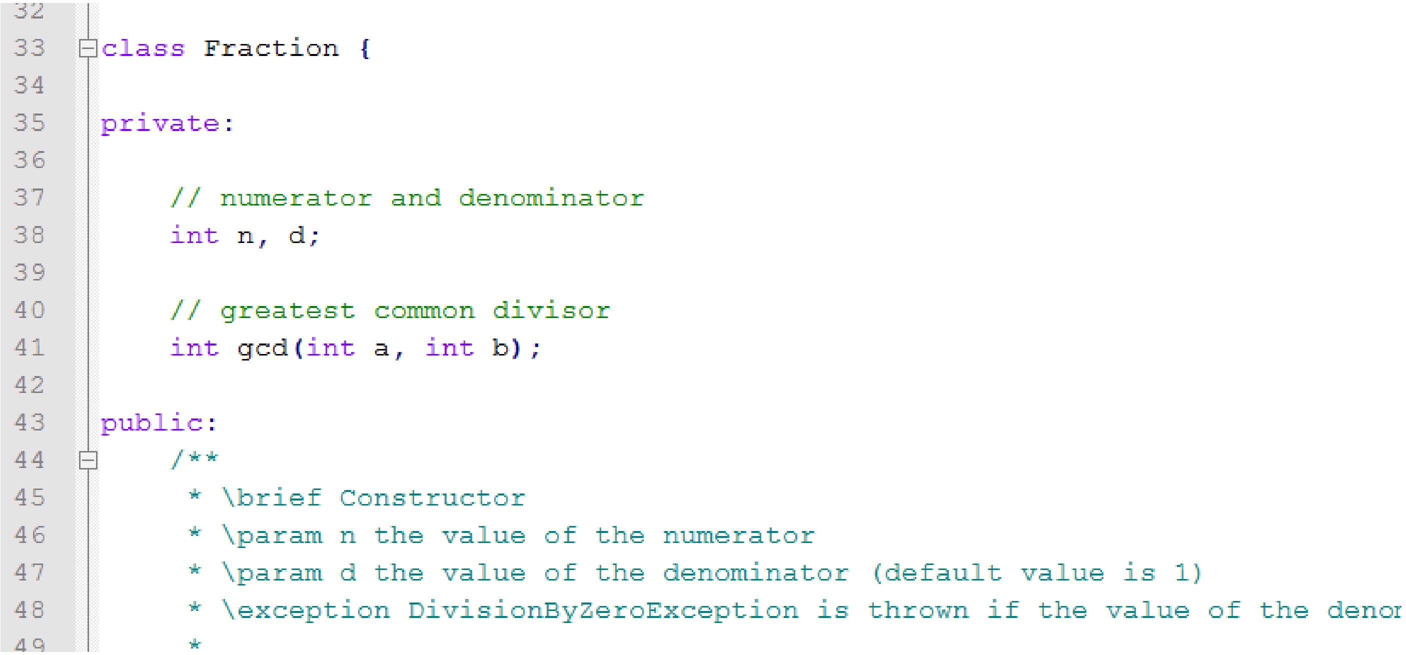
Completing the fraction class

Completing the fraction class

- Let us implement each of the public methods of the fraction class
- First we will include the code for the constructor:
  - copy and paste the code available in /work/files_tutorial/methods/constructor
  - notice that the code for computing the greatest common divisor, obtain numerator and denominator values are also included
  - it is quite straightforward to notice that at this point we need two private data members to store the numerator (*int n*) and the denominator (*int d*)

Completing the fraction class

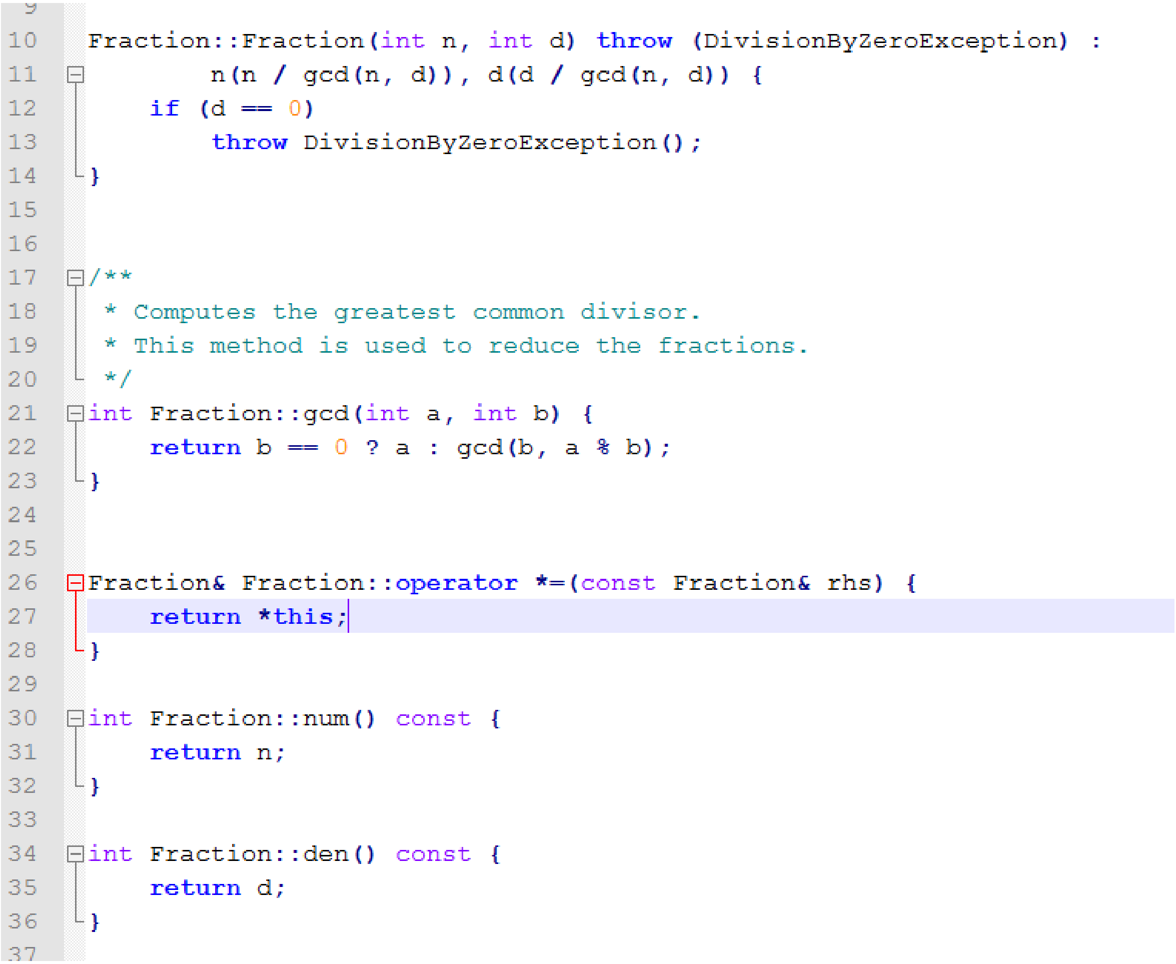
- A part of the *Fraction.hpp* should now look like this:



```
class Fraction {

private:

    // numerator and denominator
    int n, d;

    // greatest common divisor
    int gcd(int a, int b);

public:
    /**
     * \brief Constructor
     * \param n the value of the numerator
     * \param d the value of the denominator (default value is 1)
     * \exception DivisionByZeroException is thrown if the value of the denor
     *
```

Completing the fraction class

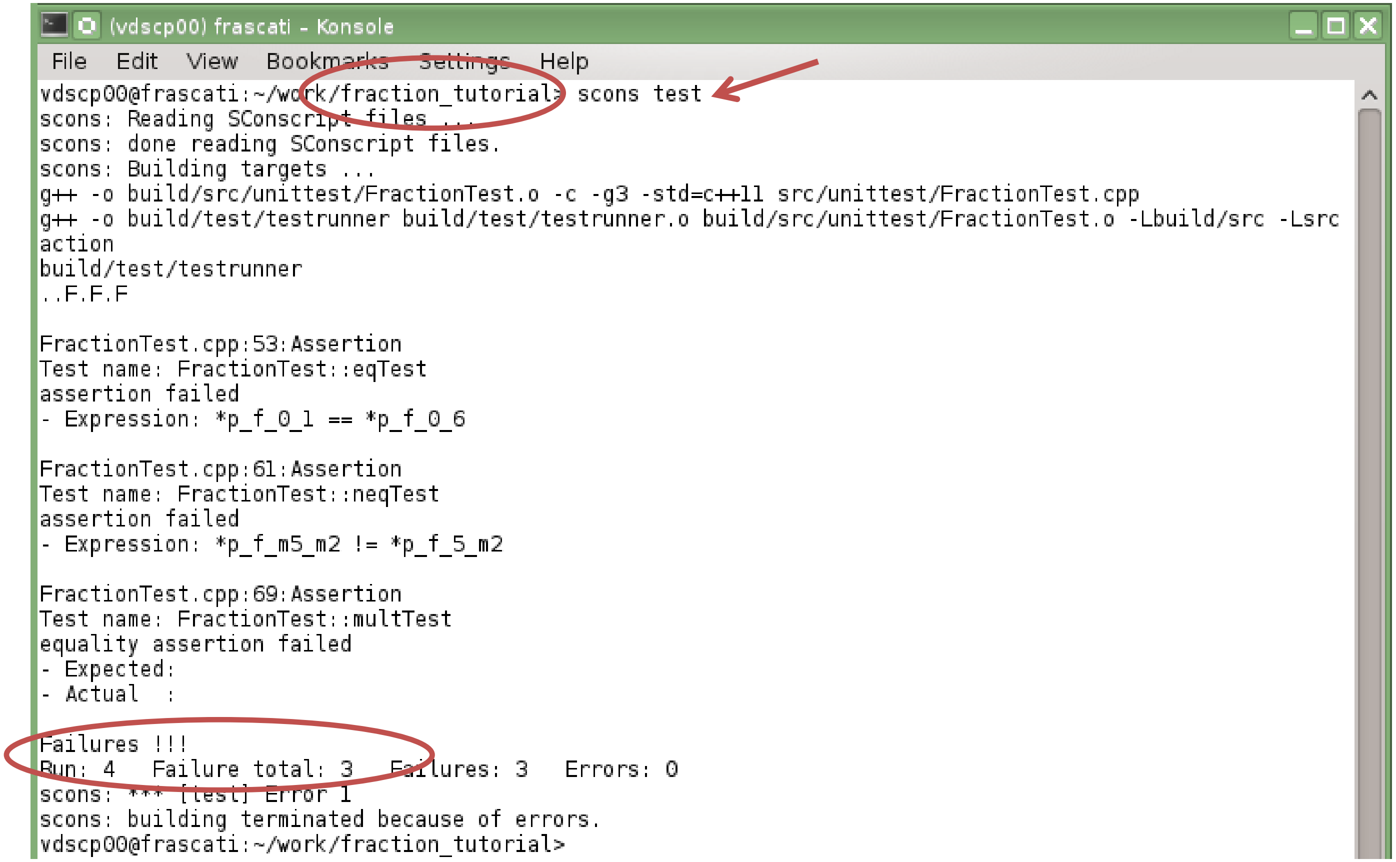
- A part of the *Fraction.cpp* should now look like this:



```
Fraction::Fraction(int n, int d) throw (DivisionByZeroException) :
    n(n / gcd(n, d)), d(d / gcd(n, d)) {
  if (d == 0)
    throw DivisionByZeroException();
}


/**
 * Computes the greatest common divisor.
 * This method is used to reduce the fractions.
 */
int Fraction::gcd(int a, int b) {
  return b == 0 ? a : gcd(b, a % b);
}


Fraction& Fraction::operator *=(const Fraction& rhs) {
  return *this;
}

int Fraction::num() const {
  return n;
}

int Fraction::den() const {
  return d;
}
```

Completing the fraction class

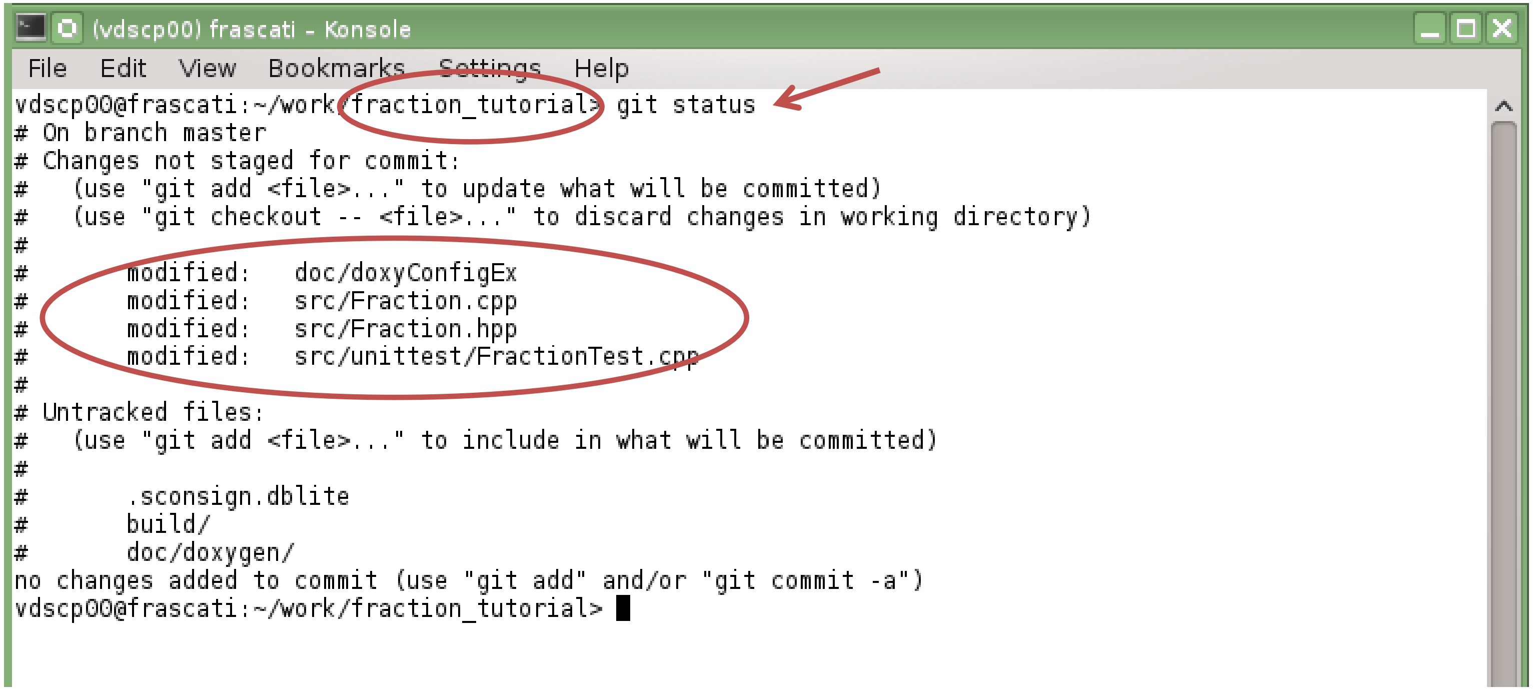
- Save, compile and run the tests using scons (or make)
- Notice that now the test for the constructor should pass



```
vdscp00@frascati:~/work/fraction_tutorial> scons test
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
g++ -o build/src/unittest/FractionTest.o -c -g3 -std=c++11 src/unittest/FractionTest.cpp
g++ -o build/test/testrunner build/test/testrunner.o build/src/unittest/FractionTest.o -Lbuild/src -Lsrc
action
build/test/testrunner
..F.F.F

FractionTest.cpp:53:Assertion
Test name: FractionTest::eqTest
assertion failed
- Expression: *p_f_0_1 == *p_f_0_6

FractionTest.cpp:61:Assertion
Test name: FractionTest::neqTest
assertion failed
- Expression: *p_f_m5_m2 != *p_f_5_m2

FractionTest.cpp:69:Assertion
Test name: FractionTest::multTest
equality assertion failed
- Expected:
- Actual  :

Failures !!!
Run: 4   Failure total: 3   Failures: 3   Errors: 0
scons: *** [test] Error 1
scons: building terminated because of errors.
vdscp00@frascati:~/work/fraction_tutorial>
```

Completing the fraction class

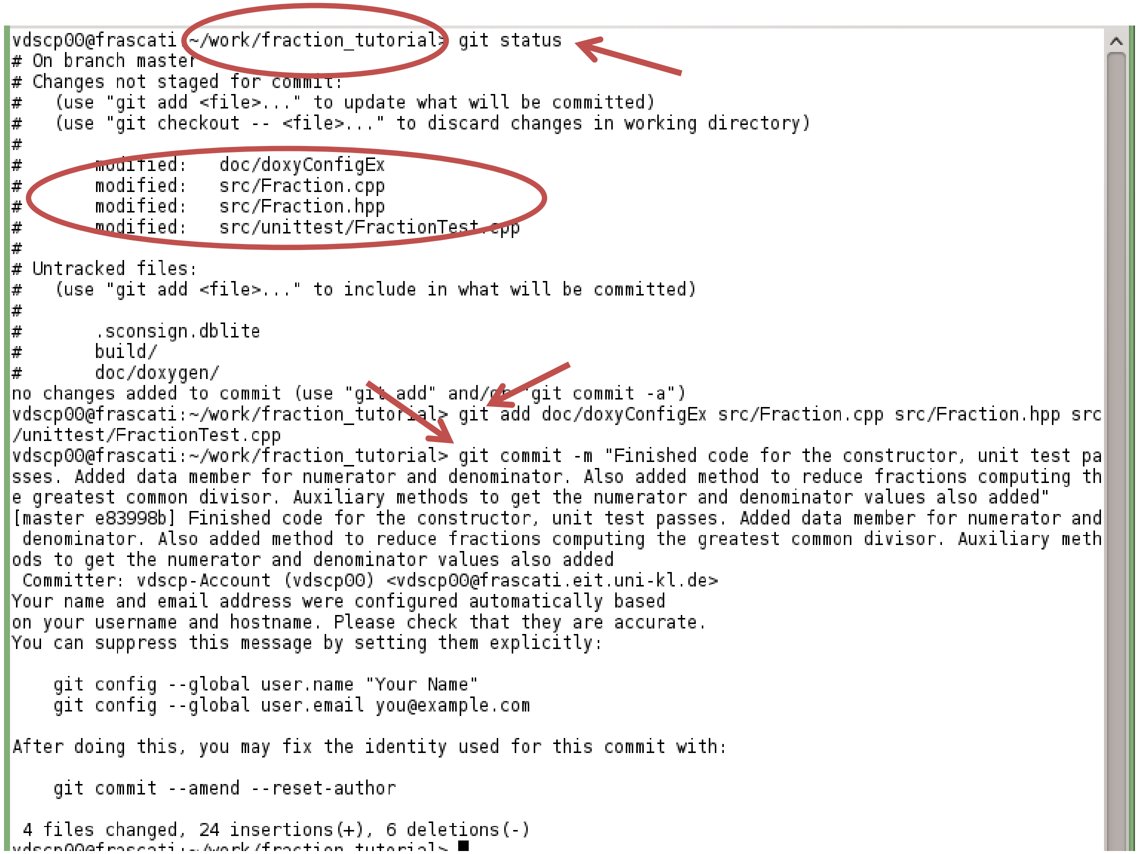
- It is time to update the local git repository
- Type git status to see that there are changes to commit (furthermore you may also push the changes to the central repository)


(vdscp00) frascati - Konsole
File Edit View Bookmarks Settings Help

```
vdscp00@frascati:~/work/fraction_tutorial> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   doc/doxyConfigEx
#       modified:   src/Fraction.cpp
#       modified:   src/Fraction.hpp
#       modified:   src/unittest/FractionTest.cpp
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .sconsign.dblite
#       build/
#       doc/doxygen/
no changes added to commit (use "git add" and/or "git commit -a")
vdscp00@frascati:~/work/fraction_tutorial> 
```

Completing the fraction class

- Add and commit the changes of your code using git



```
vdscp00@frascati:~/work/fraction_tutorial> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   doc/doxyConfigEx
#       modified:   src/Fraction.cpp
#       modified:   src/Fraction.hpp
#       modified:   src/unittest/FractionTest.cpp
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .sconsign.dblite
#       build/
#       doc/doxygen/
no changes added to commit (use "git add" and/or "git commit -a")
vdscp00@frascati:~/work/fraction_tutorial> git add doc/doxyConfigEx src/Fraction.cpp src/Fraction.hpp src
/unittest/FractionTest.cpp
vdscp00@frascati:~/work/fraction_tutorial> git commit -m "Finished code for the constructor, unit test pa
sses. Added data member for numerator and denominator. Also added method to reduce fractions computing th
e greatest common divisor. Auxiliary methods to get the numerator and denominator values also added"
[master e83998b] Finished code for the constructor, unit test passes. Added data member for numerator and
 denominator. Also added method to reduce fractions computing the greatest common divisor. Auxiliary meth
ods to get the numerator and denominator values also added
 Committer: vdscp-Account (vdscp00) <vdscp00@frascati.eit.uni-kl.de>
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:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

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

    git commit --amend --reset-author

 4 files changed, 24 insertions(+), 6 deletions(-)
vdscp00@frascati:~/work/fraction_tutorial> █
```

- If you want to save in the central repository then push the changes

Completing the fraction class

- As for the constructor follow the same steps (modify code, compile and test, commit using git) for each of the methods:
  - equality comparison method (use the *file work/files_tutorial/methods/equal*)
  - inequality comparison method (use the *file work/files_tutorial/methods/notequal*)
  - multiplication method (use the file *work/files_tutorial/methods/multiplication*)
- Note: at the end 4 new commits (one for each method) should have been done (remember to use meaningful messages in the commits)
  - before commit make sure that the corresponding unit test pases
- The next slides show how the final version of *Fraction.cpp* should look like

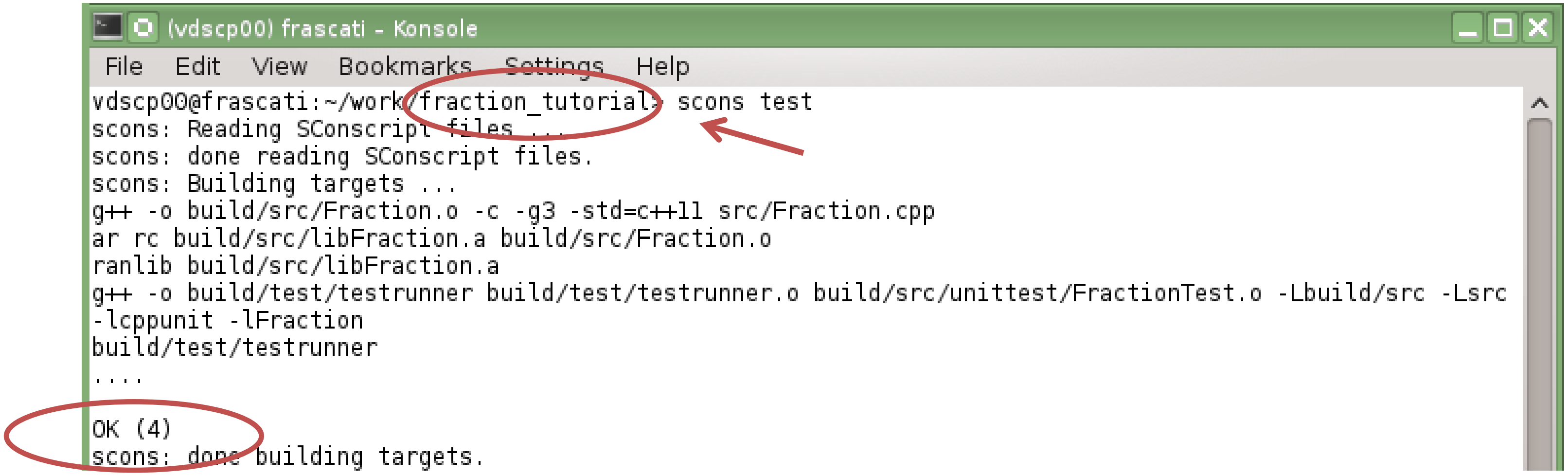
Completing the fraction class

```
/**
 * \brief Implements methods of the fraction class
 *
 * \author Carlos Villarraga
 *
 */

#include "Fraction.hpp"

Fraction::Fraction(int n, int d) throw (DivisionByZeroException) :
                        n(n / gcd(n, d)), d(d / gcd(n, d)) {
            if (d == 0)
                        throw DivisionByZeroException();
}


/**
 * Computes the greatest common divisor.
 * This method is used to reduce the fractions.
 */
int Fraction::gcd(int a, int b) {
            return b == 0 ? a : gcd(b, a % b);
}


Fraction& Fraction::operator *=(const Fraction& rhs) {
            int new_n = n * rhs.n / gcd(n * rhs.n, d * rhs.d);
            d = d * rhs.d / gcd(n * rhs.n, d * rhs.d);
            n = new_n;
            return *this;
}

int Fraction::num() const {
            return n;
}
```

```
int Fraction::den() const {
            return d;
}

Fraction operator *(const Fraction& lhs, const Fraction& rhs) {
            Fraction frc(lhs.num(), lhs.den());
            return frc *= rhs;
}

bool Fraction::operator ==(const Fraction& rhs) const {
            return this->num() == rhs.num() && this->den() == rhs.den();
}

bool Fraction::operator !=(const Fraction& rhs) const {
            return !(*this == rhs);
}

ostream& operator <<(ostream& out, const Fraction& rhs) {
            return out << rhs.num() << "/" << rhs.den();
}
```

Completing the fraction class

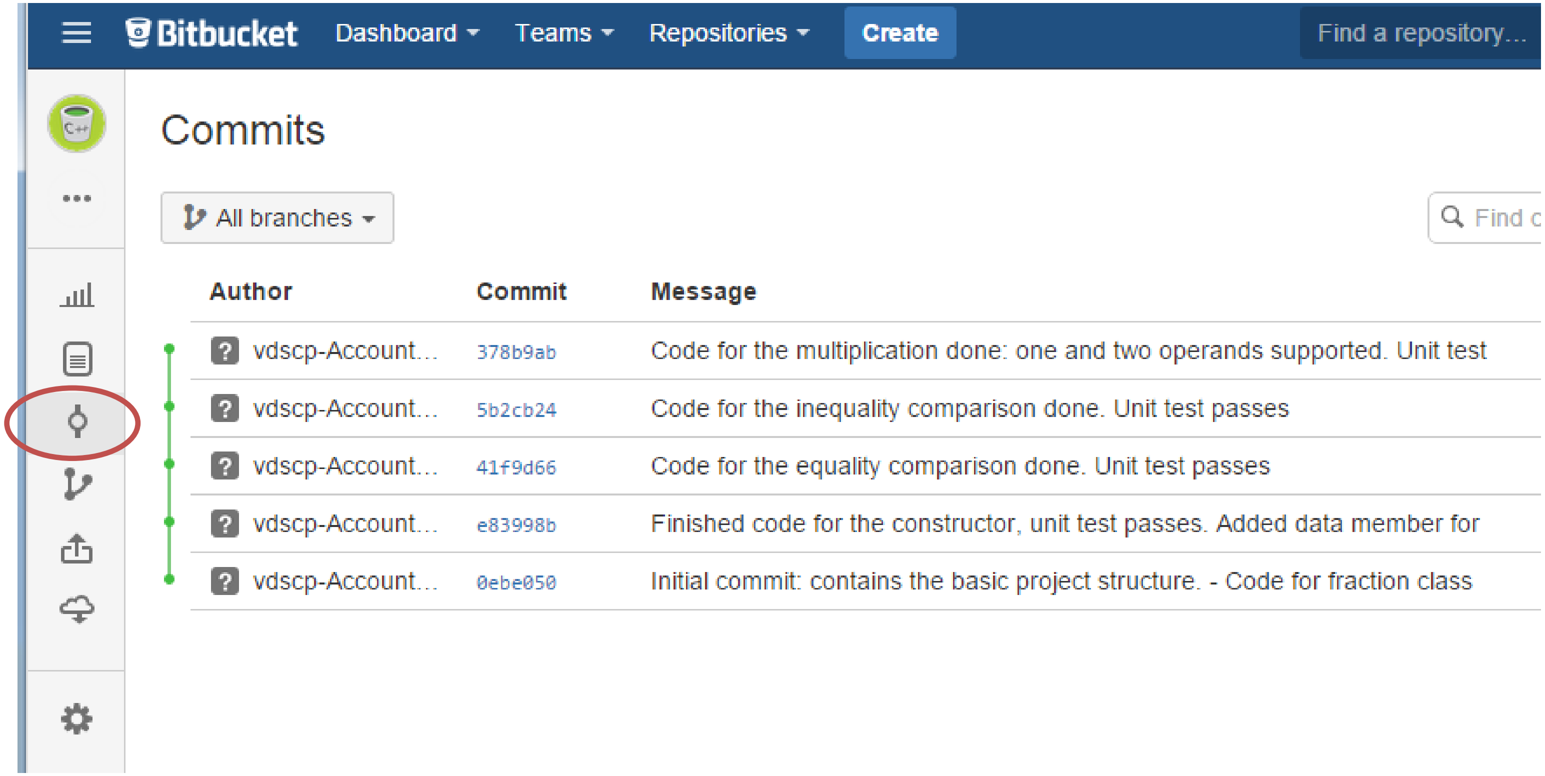
- At the end all unit tests should pass


(vdscp00) frascati - Konsole
File Edit View Bookmarks Settings Help

```
vdscp00@frascati:~/work/fraction_tutorial> scons test
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
g++ -o build/src/Fraction.o -c -g3 -std=c++11 src/Fraction.cpp
ar rc build/src/libFraction.a build/src/Fraction.o
ranlib build/src/libFraction.a
g++ -o build/test/testrunner build/test/testrunner.o build/src/unittest/FractionTest.o -Lbuild/src -Lsrc
-lcppunit -lFraction
build/test/testrunner
....

OK (4)
scons: done building targets.
```

Completing the fraction class

- Do not forget to push the final version
- Have a look in bitbucket and check that all commits are included



Commits

Author	Commit	Message
vdscp-Account…	378b9ab	Code for the multiplication done: one and two operands supported. Unit test
vdscp-Account…	5b2cb24	Code for the inequality comparison done. Unit test passes
vdscp-Account…	41f9d66	Code for the equality comparison done. Unit test passes
vdscp-Account…	e83998b	Finished code for the constructor, unit test passes. Added data member for
vdscp-Account…	0ebe050	Initial commit: contains the basic project structure. - Code for fraction class