

Software Testing

Version Control Systems

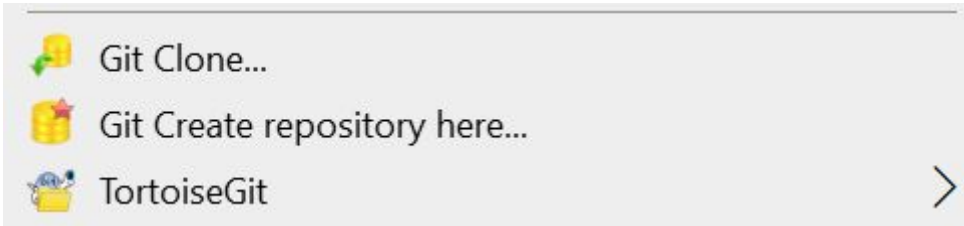
Version Control

- Version control systems are
 - repositories for software that are able to maintain multiple versions of the software.
 - This means it is able to save the previous version as well as the new version.
 - The advantage of this is that if you make a mistake and you need to revert to a previous version, this can easily be done.
- Version control can be run
 - Locally on your computer
 - Remotely on a network server
- Git is one of the most popular version control systems today.
- Seneca runs its own server at:
 - <https://github.senecacollege.ca>
 - You must use the VPN to access this

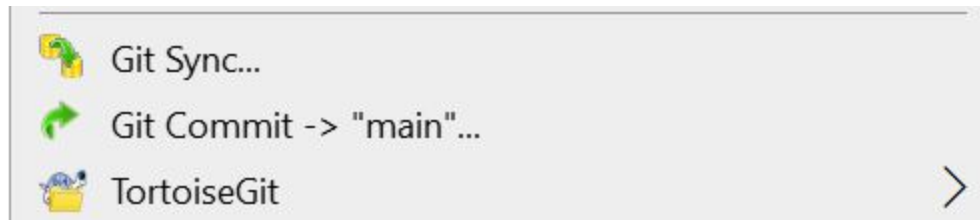
Git Clients

- A Git client is a program which communicates with either a local or remote Git repository.
- The Git client is able to
 - creating a clone of the repository,
 - add new files to the repository, and
 - retrieve information about the repository.
- There are many different git clients available.
 - TortoiseGit is integrated into the Windows file browser

TortoiseGit Menus



Non-repository

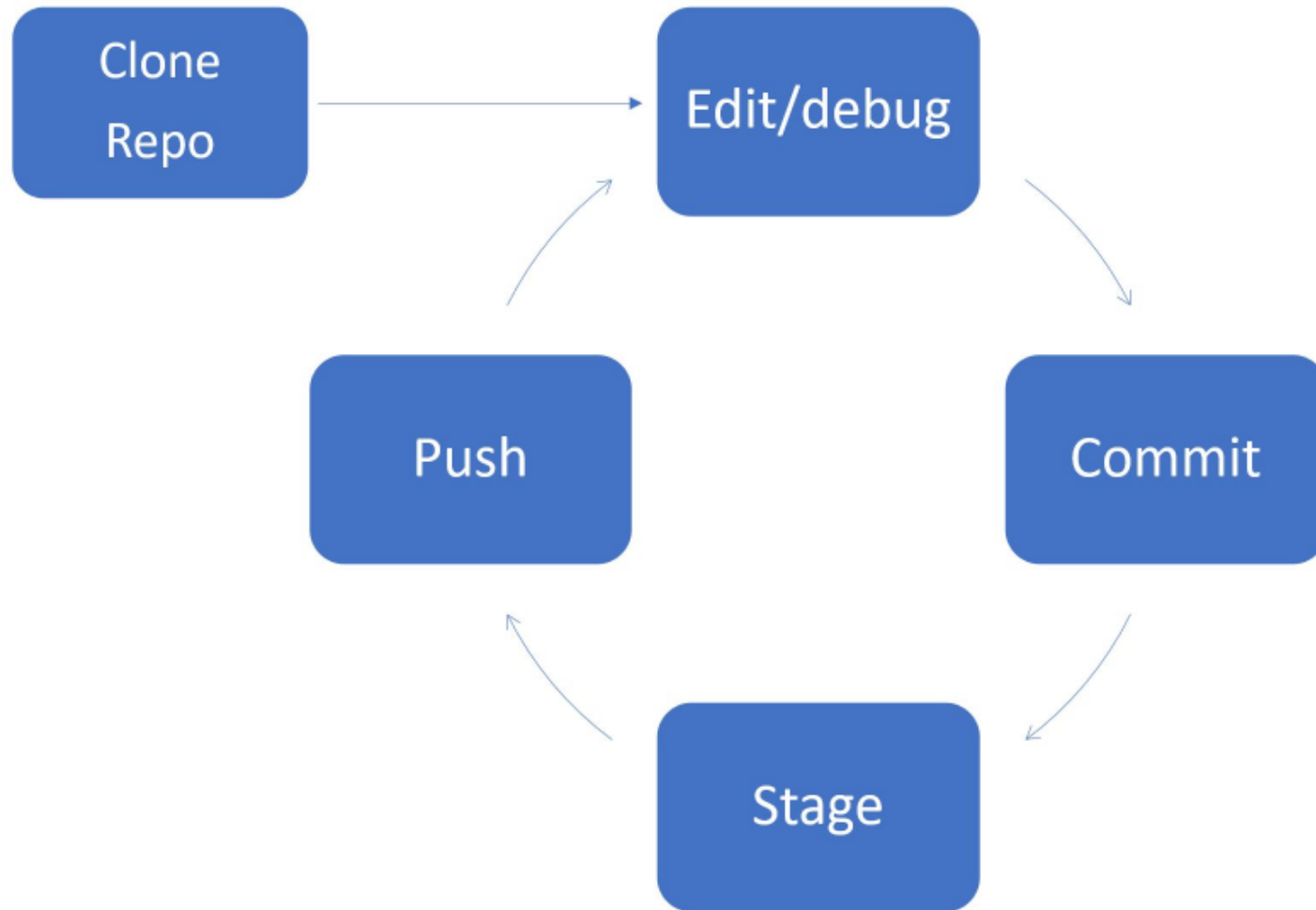


Repository

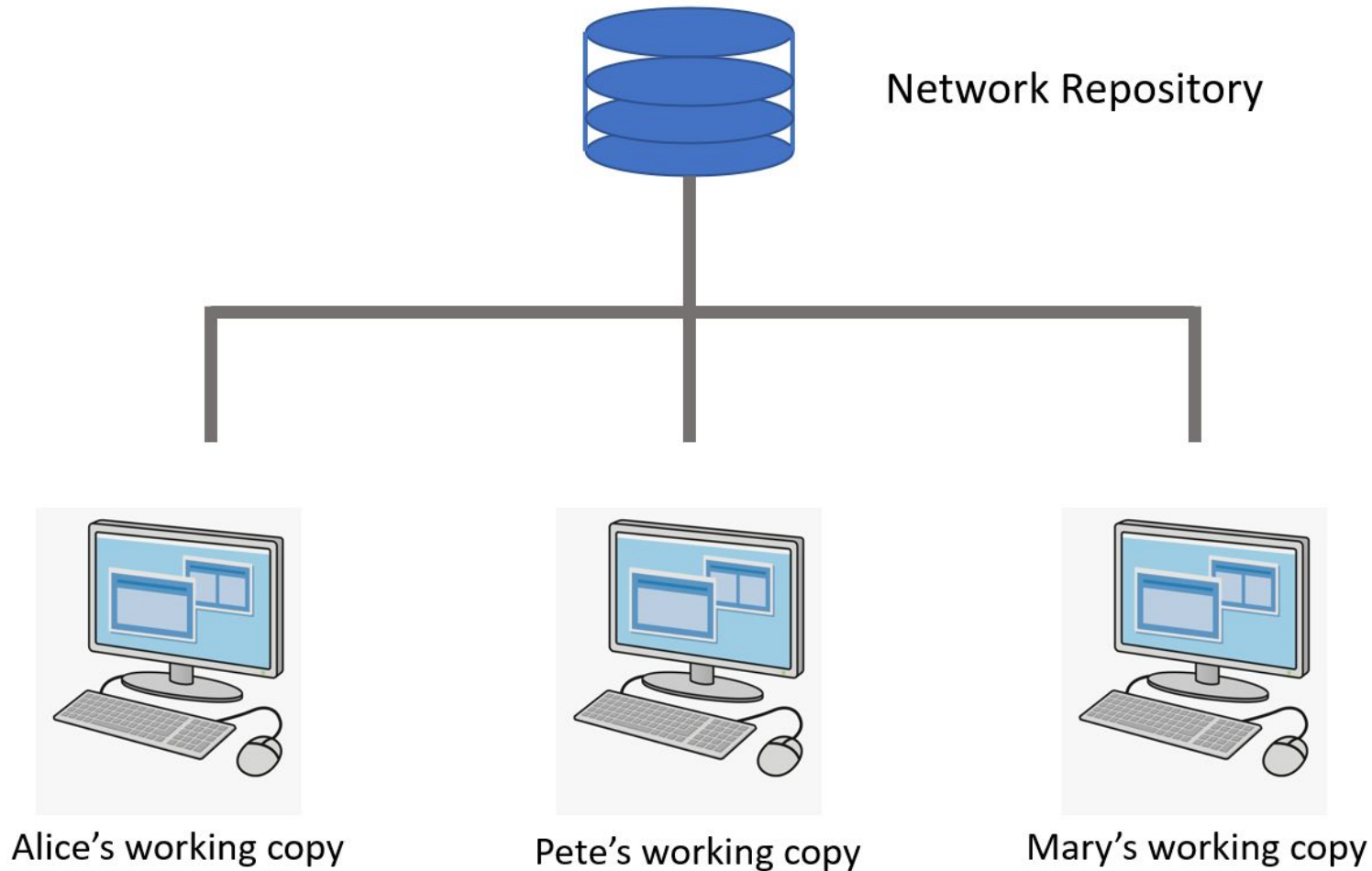
Working Copies

- You clone the remote repository to your computer
- This creates a local copy of the repository
- You make changes to the local repository
- You commit the changes to the local repository
- Stage the files you want to be in the next push
- You push the committed changes to the remote repository so others can share your work

Working Copies



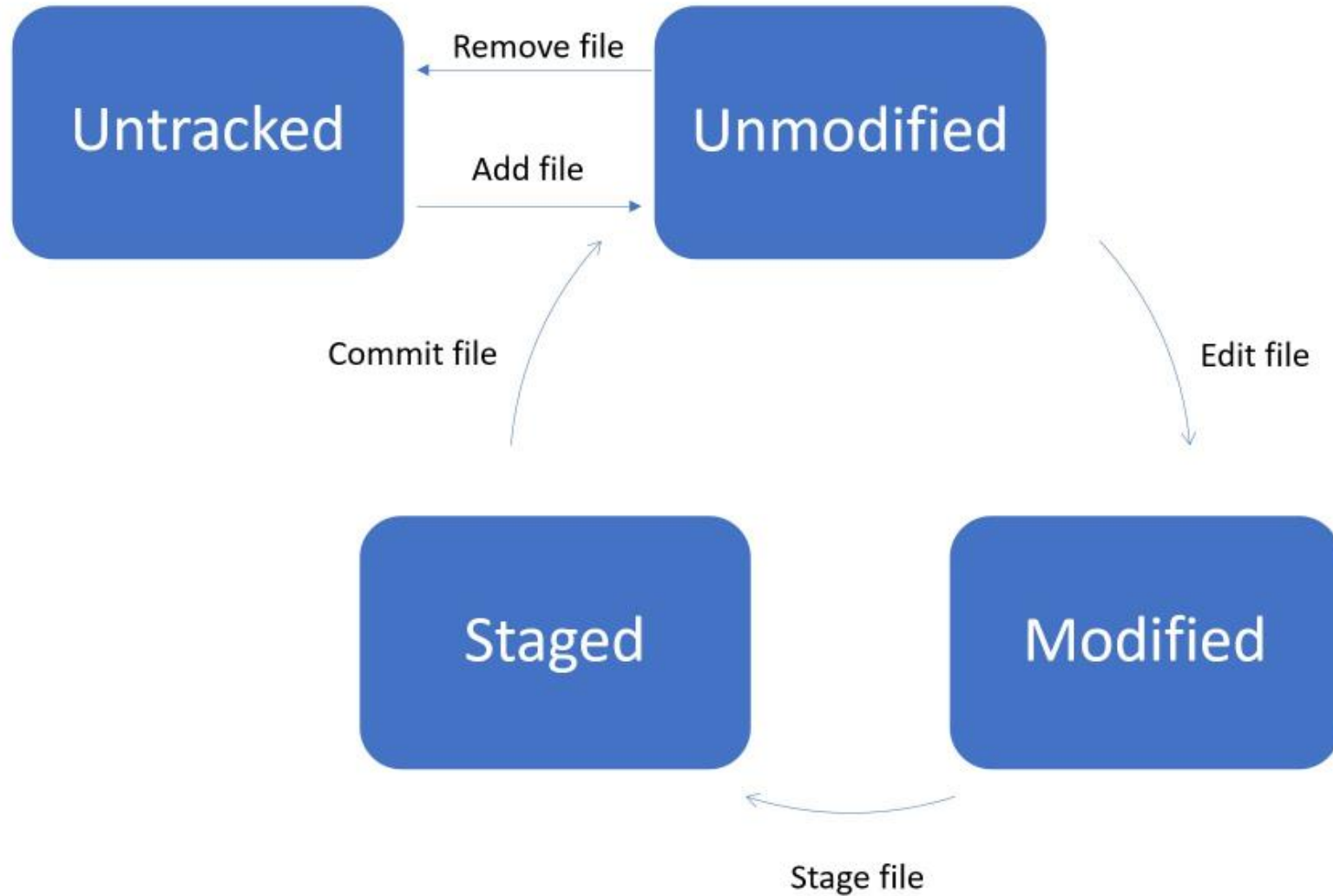
Working Copies



File Status

- Git only versions the files you ask it to
 - Projects have many intermediate file like object and executable files which can be rebuilt
- You must add a file to Git to have it tracked or versioned
- Once tracked, it tells you if the file has been modified
- You can then stage the file and commit it

File Status



Tortoise: Check for Modifications

C:\Users\user2\Documents\ptest-demo\ptest-demo - Working Tree - TortoiseGit

main

Path	Extension	Status	Lines added	Lines removed	Last Modified
SimpleVSUnitTest/.vs/SimpleVSUnitTest/v17/.suo	.suo	Modified	-	-	6/2/2022 1:04:06 PM
SimpleVSUnitTest/SimpleVSUnitTest/mathfuncs.c	.c	Modified	1	1	6/2/2022 1:04:06 PM

☒ Show unversioned files
☐ Show ignore local changes flagged files
☐ Show ignored files
☒ Show all staged files
☒ Show Whole Project

line: 1(+) 1(-) files: normal=0, non-versioned=0, modified=2, added=0, deleted=0, conflicted=0

Save unified diff Stash Commit Refresh OK

Ignoring Files

- You want Git to ignore temporary files
- List patterns for them in a .gitignore file

- *.vsidx
 - *.lock
 - *.ipch
 - *.testlog
 - *.tlog
 - *.log
 - *.VC.db
 - *.manifest
 - *.VC.db-shm
 - *.VC.db-wal
 - *.VC.db-opendb

Communicating with the Remote Repository

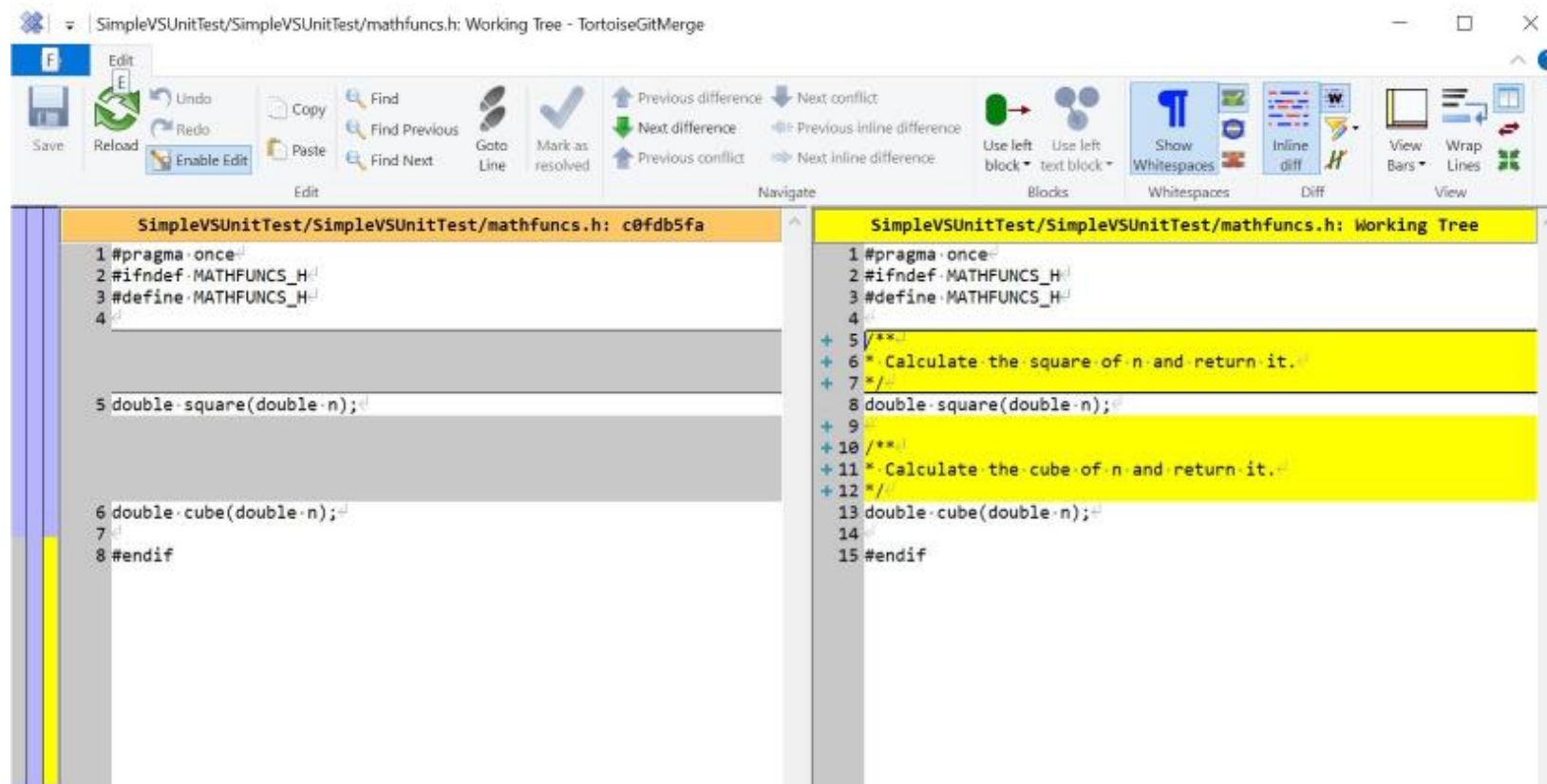
- **Clone** – creates a local copy of the repository
- **Pull** – updates your local copy with changes from the remote
- **Push** – pushes your local changes to the remote

Other Actions

- **Rename** – change the name of a file tracked by Git
- **Delete** – stop a file from being tracked by Git

Viewing Changes

Check for Modifications | Compare with Base



The screenshot displays the TortoiseGitMerge application window, titled "SimpleVSUnitTest/SimpleVSUnitTest/mathfuncs.h: Working Tree - TortoiseGitMerge". The interface is split into two panes for comparison. The left pane, labeled "SimpleVSUnitTest/SimpleVSUnitTest/mathfuncs.h: c0fdb5fa", shows the base version of the file. The right pane, labeled "SimpleVSUnitTest/SimpleVSUnitTest/mathfuncs.h: Working Tree", shows the current working tree version. The code in both panes is as follows:

```
1 #pragma once
2 #ifndef MATHFUNCS_H
3 #define MATHFUNCS_H
4
5 double square(double n);
6
7
8 double cube(double n);
9
10 #endif
```

In the working tree version (right pane), lines 5 through 12 are highlighted in yellow, indicating changes. These lines include comments and function definitions for `square` and `cube` functions. The interface includes a menu bar with "Edit", "Navigate", "Blocks", "Whitespaces", "Diff", and "View". The "Edit" menu contains options like Save, Reload, Undo, Redo, Copy, Paste, Find, Find Previous, Find Next, Goto Line, and Mark as resolved. The "Navigate" menu includes Previous difference, Next difference, Previous inline difference, Next inline difference, Previous conflict, and Next conflict. The "Blocks" menu has Use left block and Use left text block. The "Whitespaces" menu has Show Whitespaces. The "Diff" menu has Inline diff. The "View" menu has View Bars and Wrap Lines.

Conflicts – Two developers Change the Same Line

The screenshot displays the TortoiseGitMerge interface for a conflict in the file `mathfuncs.h`. The window is titled "HEAD - TortoiseGitMerge". The interface is divided into three main panes: "MERGE_HEAD (origin/HEAD, 358514cb)" on the left, "HEAD" on the right, and "Merged - mathfuncs.h" at the bottom.

MERGE_HEAD (origin/HEAD, 358514cb):

```
1 #pragma once
2 #ifndef MATHFUNCS_H
3 #define MATHFUNCS_H
4
5 // square and cube functions
6 double square(double n);
7
8
9 double cube(double n);
10
11 #endif
```

HEAD:

```
1 #pragma once
2 #ifndef MATHFUNCS_H
3 #define MATHFUNCS_H
4
5 /**
6  * Calculate the square of n and return it.
7  */
8 double square(double n);
9
10 /**
11  * Calculate the cube of n and return it.
12  */
13 double cube(double n);
14
15 #endif
```

Merged - mathfuncs.h:

```
1 #pragma once
2 #ifndef MATHFUNCS_H
3 #define MATHFUNCS_H
4
5 // square and cube functions
6 double square(double n);
7
8
9 double cube(double n);
10
11 #endif
```

The conflict is highlighted with colored bars: a red bar for the conflicting line in the MERGE_HEAD view, a yellow bar for the conflicting line in the HEAD view, and a red bar for the conflicting line in the Merged view. The Merged view shows the result of the merge, with the conflicting line from the HEAD view being used.

The status bar at the bottom indicates the current view settings: "Left View: ASCII • CRLF • Tab 4 • - 1 / + 1", "Right View: ASCII • CRLF • Tab 4 • - 1 / + 7", and "Bottom View: ASCII • CRLF • Tab 4 • - 1 / + 4 / 13".