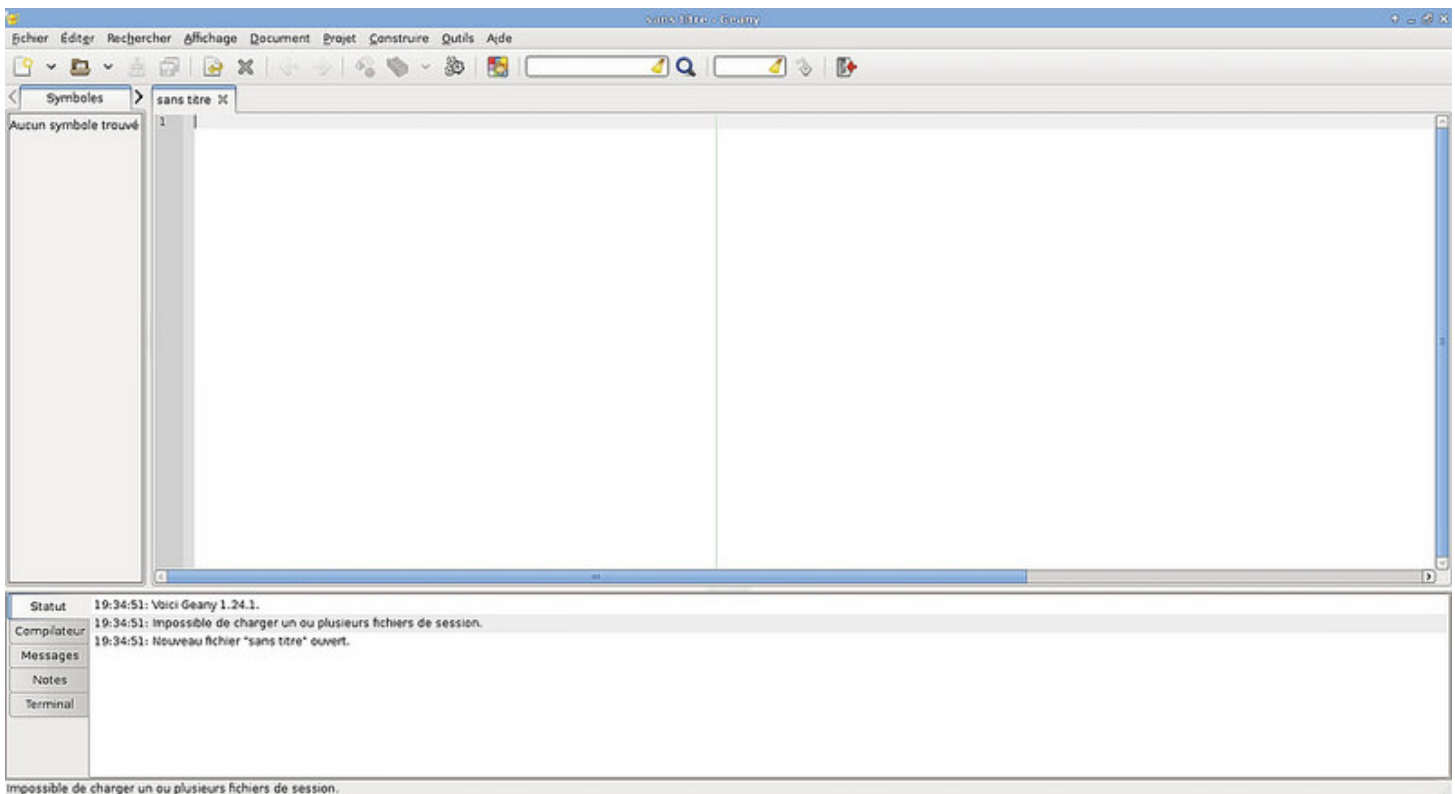


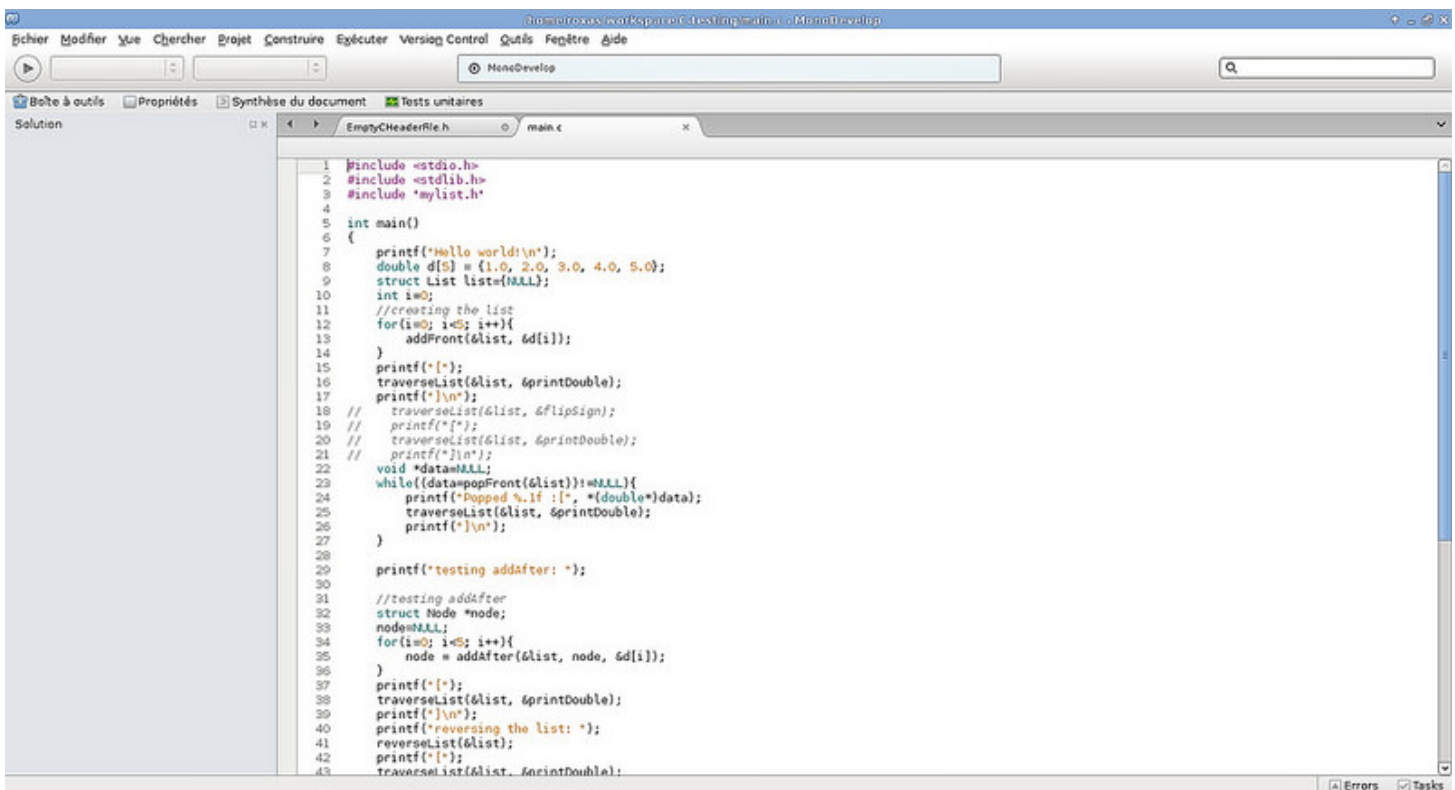
I know that I said only "real out-of-the-box IDE" and not a text editor pumped with plugins, but Eclipse is a "real out-of-the-box IDE." It's just that Eclipse needs a little plugin (or a variant) to code in C. So I technically did not contradict myself. And it would have been impossible to make an IDE list without mentioning the behemoth that is Eclipse. Like it or not, Eclipse remains a great tool to code in Java. And thanks to the CDT Project, it is possible to program in C/C++ too. You will benefit from all the power of Eclipse and its traditional features like word completion, code outline, code generator, and advanced refactoring. What it lacks in my opinion is the lightness of Code::Blocks. It is still very heavy and takes time to load. But if your machine can take it, or if you are a hardcore Eclipse fan, it is a very safe option.

### 3. Geany



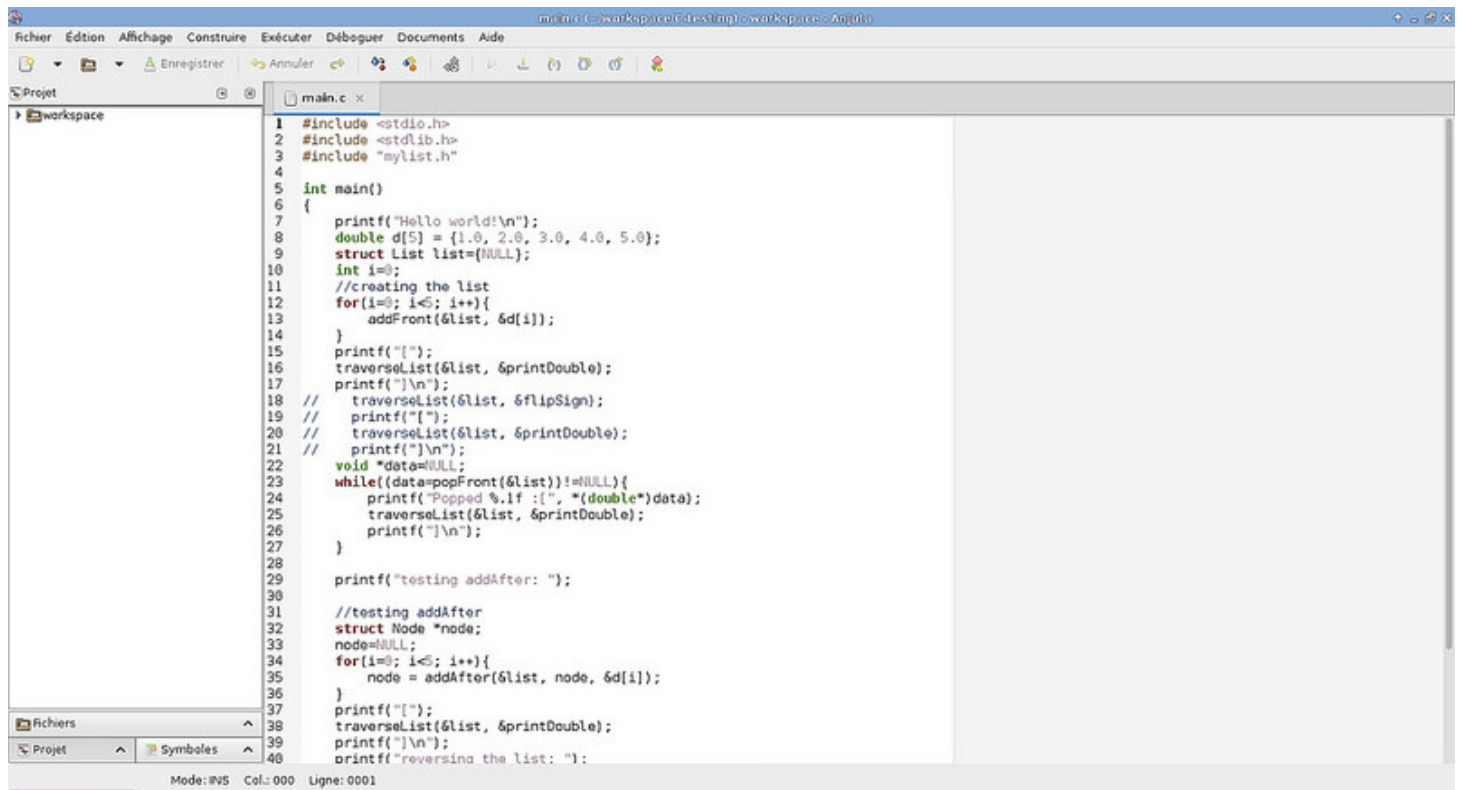
With a lot less features but a lot more flexibility, Geany is at the opposite of Eclipse. But what it lacks (like a debugger for example), Geany makes it up with nice little features: a space for note taking, creation from template, code outline, customizable shortcuts, and plugins management. Geany is still closer to an extensive text editor than an IDE here. However I keep it in the list for its lightness and its well designed interface.

## 4. MonoDevelop



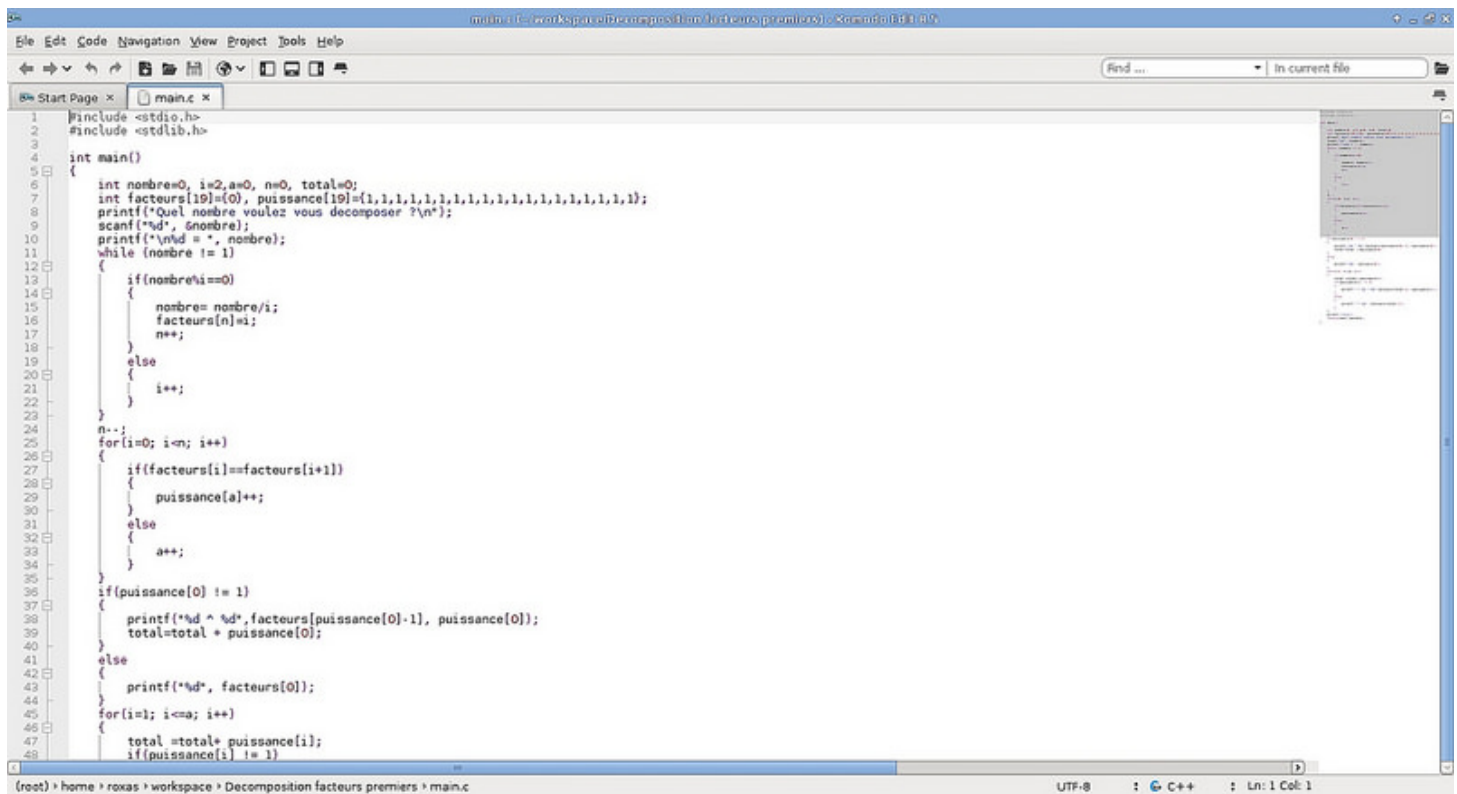
Another monster to add to the list, [MonoDevelop](#) has a very unique feel derived from its look and interface. I personally love its project management and its integrated version control system. The plugin system is also pretty amazing. But for some reason, all the options and the support for all kind of programming languages make it feel a bit overwhelming to me. It remains a great tool that I used many times in the past, but just not my number one when dealing with "simplistic" C.

## 5. Anjuta



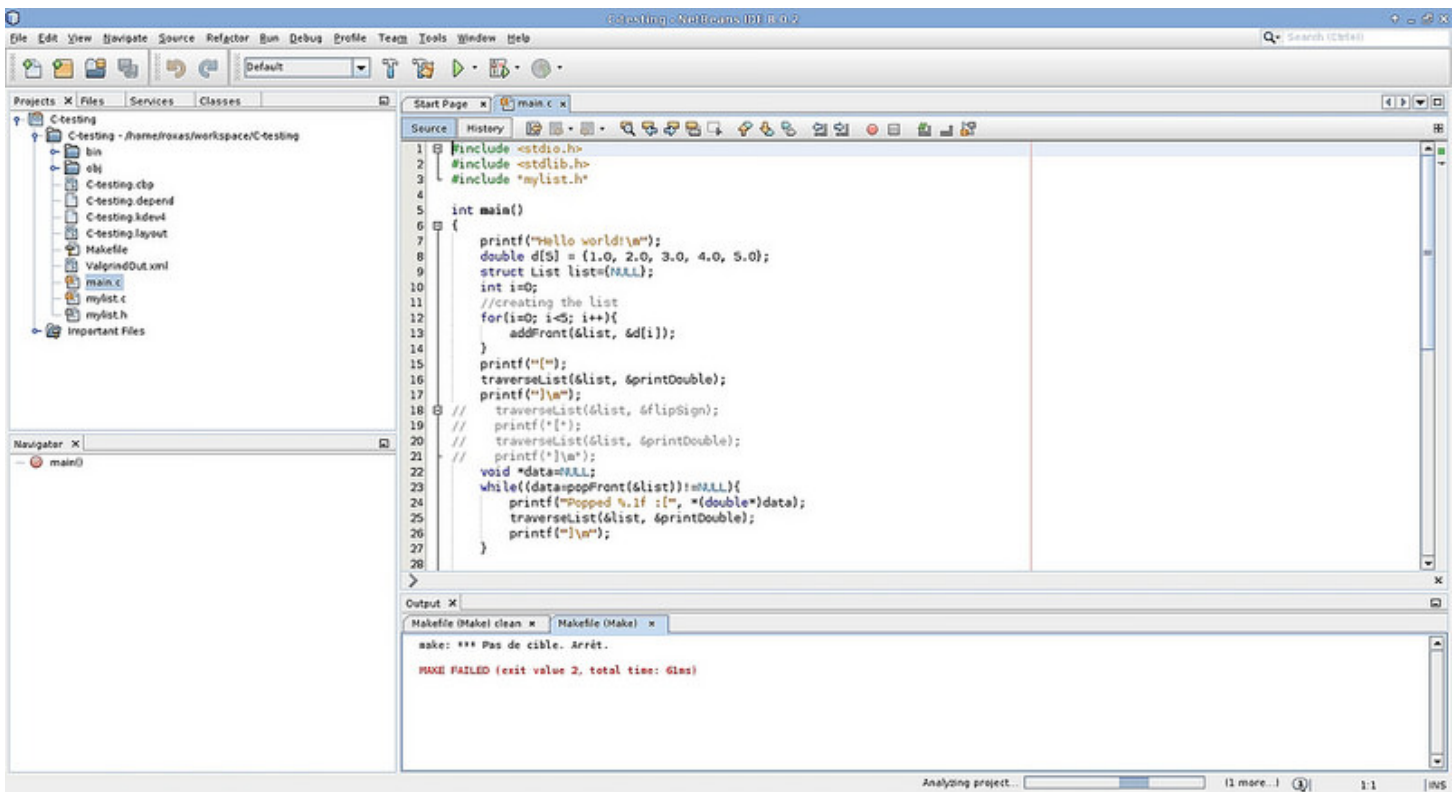
With a very strong "GNOME feeling" attached to it, [Anjuta](#)'s appearance is a hit or miss. I tend to see it as an advanced version of Geany with a debugger included, but the interface is actually a lot more elaborate. I do enjoy the tab system to switch between the project, folders, and code outline view. I would have liked maybe a bit more shortcuts to move around in a file. However, it is a good tool, and offers outstanding compilation and build options, which can support the most specific needs.

## 6. Komodo Edit



I was not very familiar with Komodo Edit, but after trying it a few days, it surprised me with many many good things. First, the tab-based navigation is always appreciable. Then the fancy looking code outline reminds me a lot of Sublime Text. Furthermore, the macro system and the file comparator make Komodo Edit very practical. Its plugin library makes it almost perfect. "Almost" because I do not find the shortcuts as nice as in other IDEs. Also, I would enjoy more specific C/C++ tools, and this is typically the flaw of general IDEs. Yet, very enjoyable software.

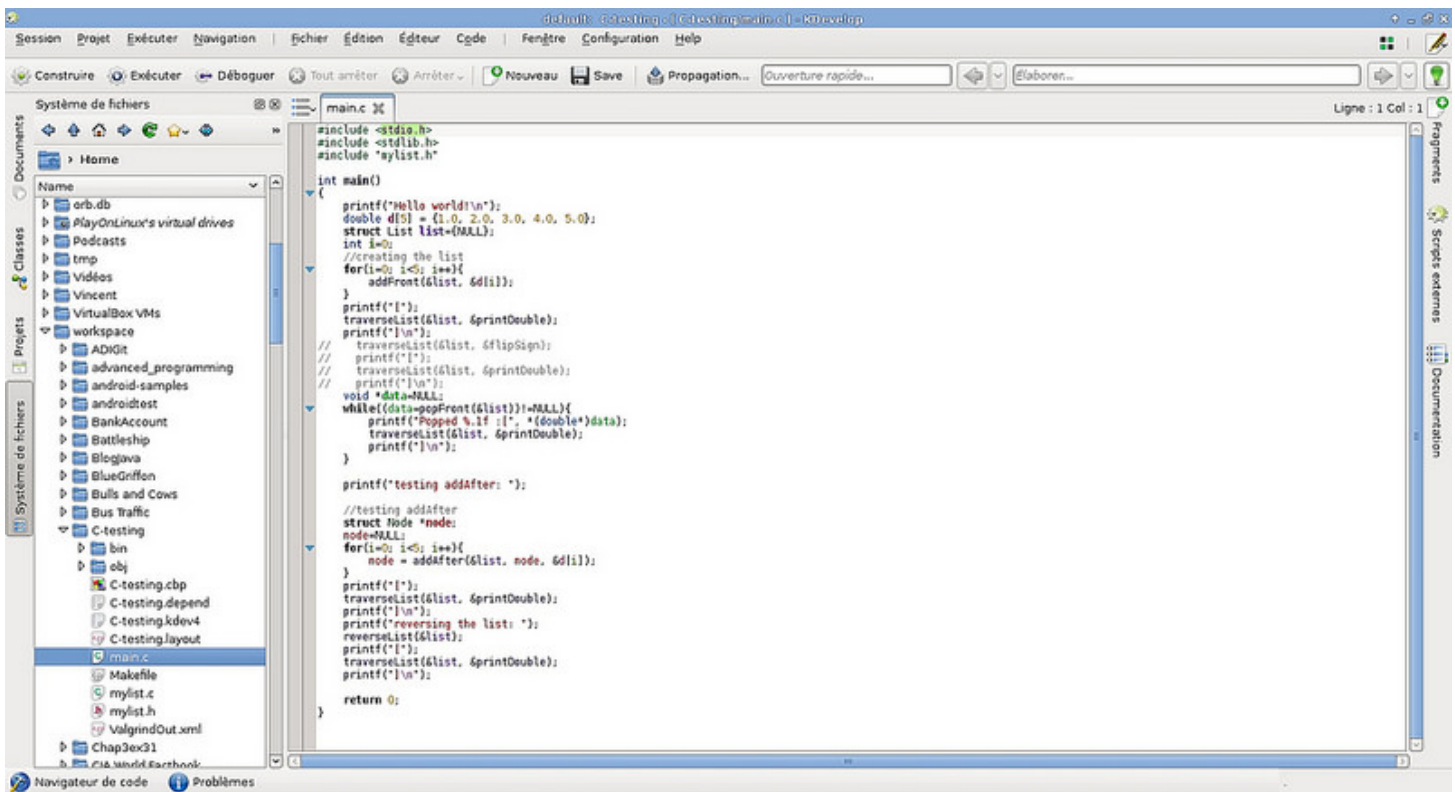
## 7. NetBeans



Just like Eclipse, impossible to avoid this beast. With navigation via tabs, project management, code outline, change history tracking, and a plethora of tools, NetBeans might be the most complete IDE out there. I could list for half a page all of its amazing features. But that will tip you off too easily about its main disadvantage, it might be too big. As great as it is, I prefer plugin based software because I doubt that anyone will need both Git and Mercurial integration for the same project. Call me crazy. But if you have the patience to master all of its options, you will be pretty much become the master of IDEs everywhere.

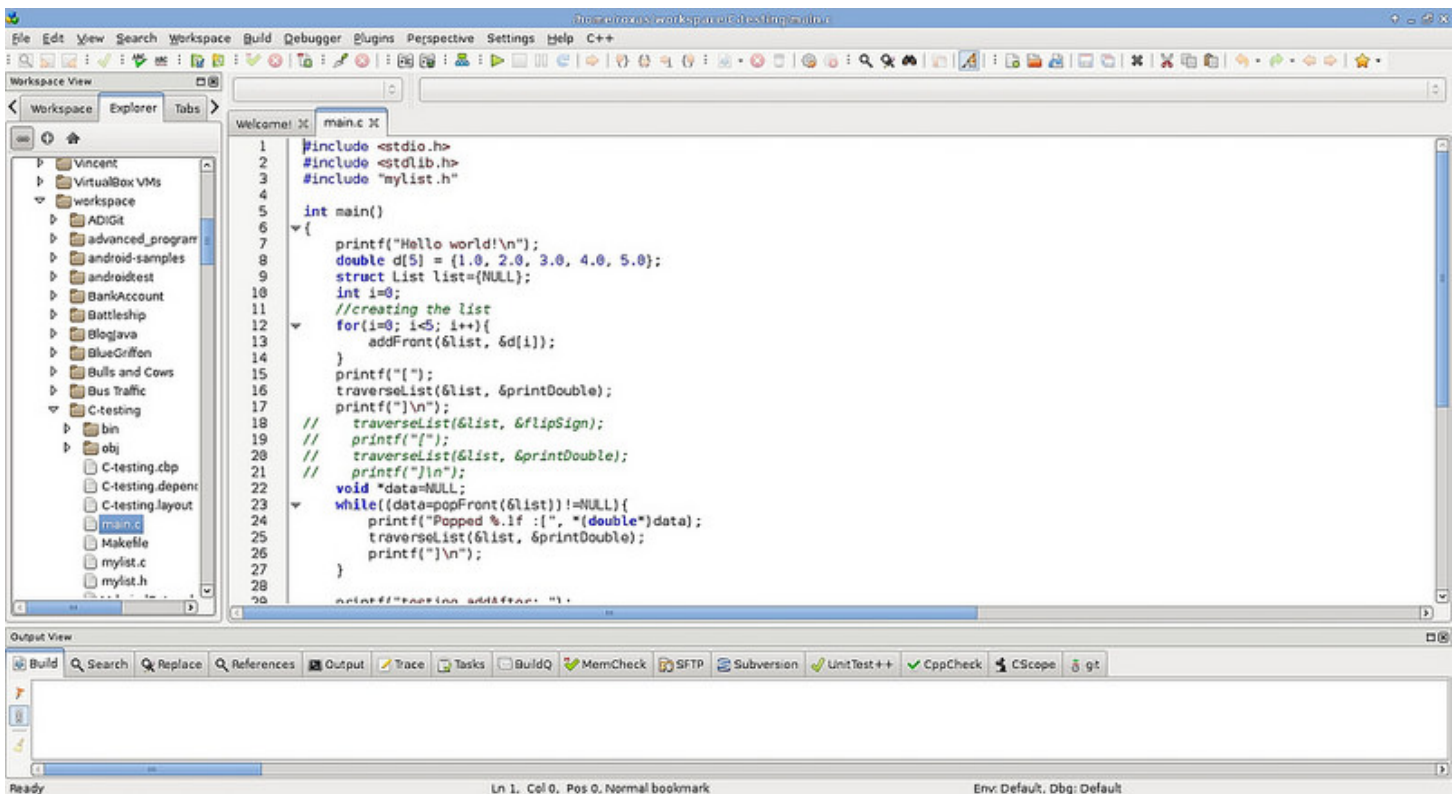
## 8. KDevelop





For all KDE fans out there, [KDevelop](#) might be the answer to your prayers. With a lot of configuration options, KDevelop is yours if you manage to seize it. Call me superficial but I never really got past the interface. But it's too bad for me as the editor itself packs quite a punch with a lot of navigation options and customizable shortcuts. The debugger is also very advanced and will take a bit of practice to master. However, this patience will be rewarded with this very flexible IDE's full power. And it gets special credits for its amazing embedded documentation.

## 9. CodeLite



Finally, last but not least, CodeLite shows that you can take a traditional formula and still get something with its own feeling attached to it. If the interface really reminded me of Code::Blocks and Anjuta at first, I was just blown away by the extensive plugin library. Whether you want to diff a file, insert a copyright block, define an abbreviation, or push your work on Git, there is a plugin for you. If I had to nitpick, I would say that it lacks a few navigation shortcuts for my taste, but that's really it.

To conclude, I hope that this list had you discover new IDEs for coding in your favorite language. While Code::Blocks remains my favorite, it has some serious challengers. Also we are far from covering all the ways to code in C/C++ using an IDE on Linux. So if you have another one to propose, let us know in the comments. Also if you would like me to cover IDEs for a different language next, also let us know in the comment section.



Do you want to receive **Linux FAQs, detailed tutorials and tips** published at Xmodulo? Enter your email address below, and we will deliver our Linux posts straight to your email box, for free. Delivery powered by Google Feedburner.

Your email address



Did you find this tutorial helpful? Then please be generous and support Xmodulo!





I am a Linux aficionado from France. After trying multiple distributions, I finally settled for Archlinux. But I am always trying to improve my system by stacking up tips and tricks.



- [How to set up C/C++ development environment in Eclipse](#)
- [How to install Eclipse on Linux](#)
- [How to fix Android SDK Content Loader stuck at 0% in Eclipse](#)
- [How to turn Vim into a full-fledged IDE](#)

Tags: c, eclipse, ide