# Working with Code

UIC Linux Users Group

September 27, 2013

outline
**editors**
compiling
debugging
resources

**what you use**
nano/pico
vim
eclipse

## beginner logic

- ▶ concept: programing is hard and scary. we should give beginers tools with limited capabilites
- ▶ problem: beginners don't want to stay beginners
- ▶ learn to use a decent IDE with a decent debugger, decent code prediction, syntax highlighting as soon as you are able

outline
**editors**
compiling
debugging
resources

what you use
**nano/pico**
vim
eclipse

## nano

- ▶ familiar interface
- ▶ included in almost every linux distribution

outline
**editors**
compiling
debugging
resources

what you use
**nano/pico**
vim
eclipse

## example

nano -w hello.txt

## vim

- ▶ Vim is an advanced text editor that seeks to provide the power of the de-facto Unix editor 'Vi', with a more complete feature set.
- ▶ Also included in almost every linux distribution

outline
**editors**
compiling
debugging
resources

what you use
nano/pico
**vim**
eclipse

## vim basics

- ▶ arrows/mouse : move around the file
- ▶ i : insert a : append
- ▶ d : delete/cut
- ▶ y : yank/copy
- ▶ wq : save and quit
- ▶ w : save
- ▶ u : undo
- ▶ redo : ctrl+r

outline
**editors**
compiling
debugging
resources

what you use
nano/pico
**vim**
eclipse

## more vim

- ▶ sp : splits the window
- ▶ m[letter] : assigns a macro
- ▶ v : visual mode
- ▶ ctrl+v : visual block
- ▶ gD : takes you to a variable/method declaration
- ▶ ctrl+n : auto-completion

outline
**editors**
compiling
debugging
resources

what you use
nano/pico
**vim**
eclipse

## configuring vim

- ▶ set autoindent
- ▶ set smartindent
- ▶ set number : displays line numbers in the left margin

outline
**editors**
compiling
debugging
resources

what you use
nano/pico
vim
**eclipse**

## eclipse

- ▶ advanced and powerful IDE (Integrated Development Enviornment)
- ▶ Primarily used for Java development, however can be used for C/C++, PHP, Android development as well

outline
editors
**compiling**
debugging
resources

g++
javac

# g++

- ▶ g++ myprogram.cpp -o myprogram
- ▶ ./myprogram

outline
editors
**compiling**
debugging
resources

g++
**javac**

## javac

- ▶ javac myprogram.java
- ▶ java -cp myprogram

## introduction

- ▶ debugging is the process of identifying the root cause of an error and correcting it
- ▶ debugging can take up to 50 percent of development time and can be the hardest part
- ▶ forces you to read and understand your code

outline
editors
compiling
**debugging**
resources

introduction
**tools**
tips

## tools

- ▶ gdb: (GNU Project Debugger)
- ▶ ddd: graphical front end for gdb
- ▶ compiler!!
- ▶ eclipse has a very good debugger for your java code

outline
editors
compiling
**debugging**
resources

introduction
tools
**tips**

## tips

- ▶ try using a binary search algorithm to find the source of the error
- ▶ talk to someone about your problem a.k.a. confessional debugging
- ▶ take a break from the problem (go for a walk, get some coffee, etc)
- ▶ make sure to save the original source (version control?)
- ▶ well formatted and organized code makes debugging a lot easier

outline
editors
compiling
debugging
**resources**

resources

## resources

- download eclipse: http://www.eclipse.org/downloads/
- VIM cheatsheet: http://www.zalas.eu/uploads/wp/2010/05/vi-vim-cheat-sheet.gif
- Guide to Faster, Less Frustrating Debugging: http://heather.cs.ucdavis.edu/ matloff/UnixAndC/CLanguage/Debug.html
- Guide to Makefiles: http://www.delorie.com/djgpp/doc/ug/larger/makefiles.html