# Week 1

# Programming in Linux

Source from Dr. Norazah Yusof (Prof. Madya ), Lecture Note

edited by Dr. Dayang Norhayati Abang Jawawi (Prof. Madya ), UTM *dayang@utm.my*

## How to Create a First C++ Program in Linux

- Normally Linux programmers type their programs by using the vi editor & Emacs editor .
- You can also you any other GUI or text editors (choose from “Applications”)

## Text Editor

- Like a word processor on a personal computer, except that it does not apply formatting styles (bold, italics, different fonts etc.).
- Crucial tools for using Unix / Linux
- Main editors:
  - vi
  - emacs
  - gedit

## 1.3

## Introduction to vi Editor

# vi Editor

- The vi editor is a screen-based editor used by many Unix users.
- The vi editor has powerful features to aid programmers, but many beginning users avoid using vi because the different features overwhelm them.
- vi editor is a text editor program that can be used to create and modify text files.
  - Simple and small

# vi editor

- Vi is a modal editor, it operates in either:
  - *insert mode* (for editing text where typed text becomes part of the document) or
  - *normal mode* (for giving commands
  - where keystrokes are interpreted as commands that control the edit session).

- Typing i while in normal mode switches the editor to insert mode.
- How the i keystroke is processed depends on the editor mode. From insert mode, pressing the escape key switches the editor back to normal mode.

# Vi Editor - Exercise

- Edit file using vi editor
- Lab 1
  - Exercise 1 (1.3)
  - Question 2
  - Page 18

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

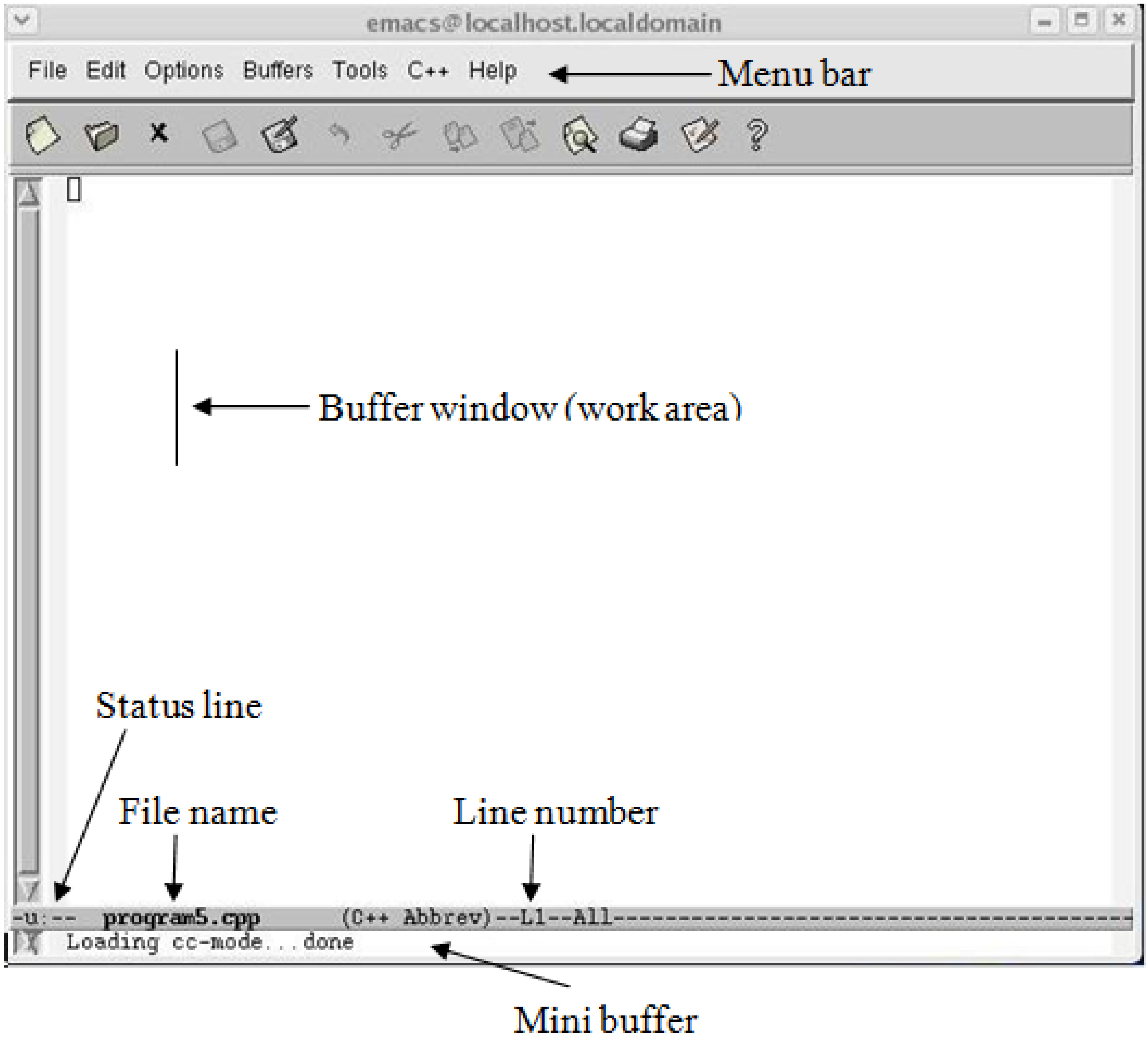
## 1.4

## Introduction to Emacs Editor

## EMACS Editor

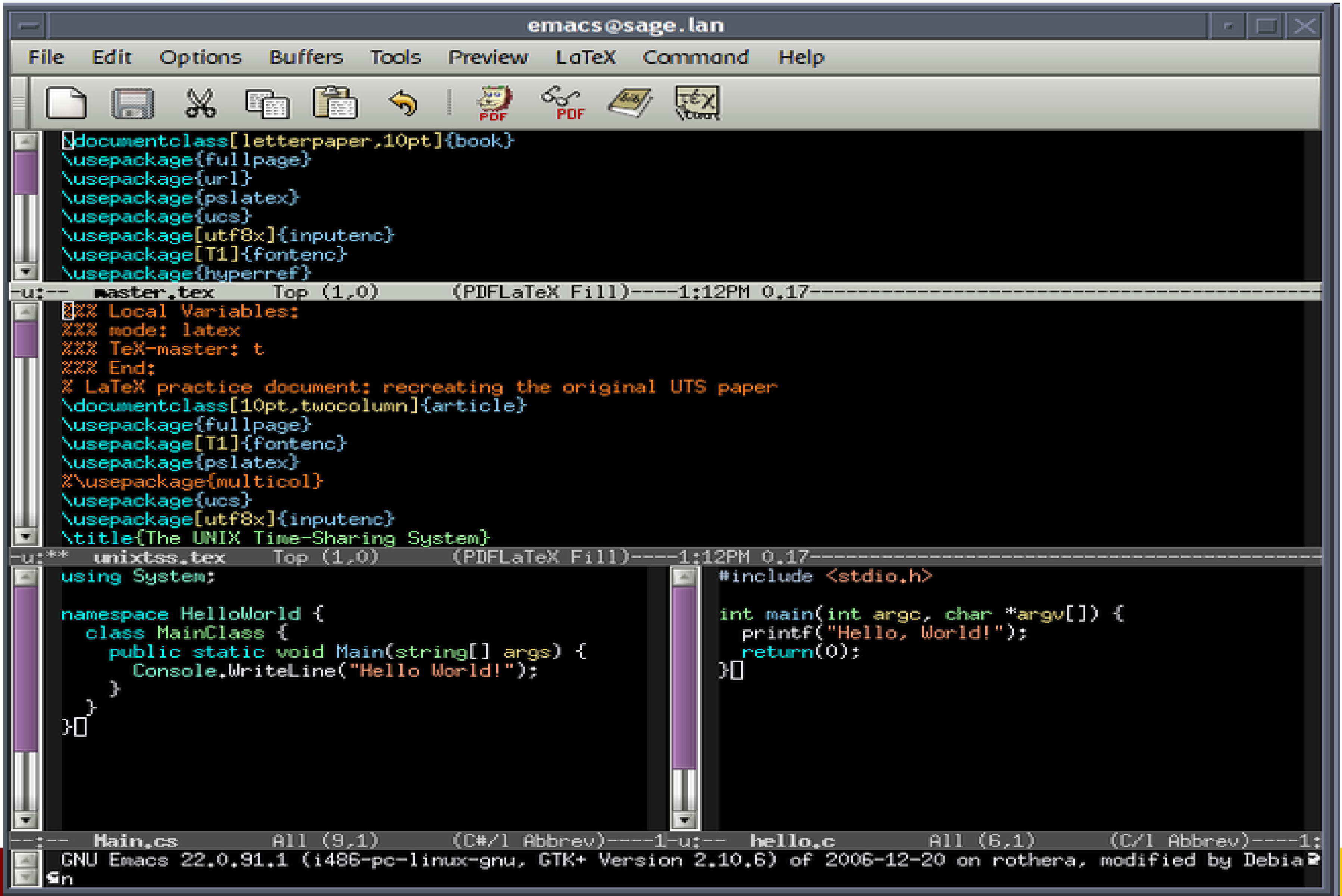
- A Unix based text editor
- Run using graphical user interface & text based user interface
- Like vi, EMACS is a screen editor.
- Unlike vi, EMACS is not an insertion mode editor.

# EMACS Editor


emacs@localhost.localdomain
File Edit Options Buffers Tools C++ Help
Menu bar
Buffer window (work area)
Status line
File name
Line number
Mini buffer

```
-u:--  program5.cpp     (C++ Abbrev)--L1--All----------------------------------------
 Loading cc-mode...done
```

# EMACS Editor



```
\documentclass[letterpaper,10pt]{book}
\usepackage{fullpage}
\usepackage{url}
\usepackage{pslatex}
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{hyperref}
-u:--  master.tex     Top (1,0)      (PDFLaTeX Fill)----1:12PM 0.17----------
%%% Local Variables:
%%% mode: latex
%%% TeX-master: t
%%% End:
% LaTeX practice document: recreating the original UTS paper
\documentclass[10pt,twocolumn]{article}
\usepackage{fullpage}
\usepackage[T1]{fontenc}
\usepackage{pslatex}
%\usepackage{multicol}
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
\title{The UNIX Time-Sharing System}
-u:**  unixtss.tex    Top (1,0)      (PDFLaTeX Fill)----1:12PM 0.17----------
```

```
using System;

namespace HelloWorld {
  class MainClass {
    public static void Main(string[] args) {
      Console.WriteLine("Hello World!");
    }
  }
}
--:--  Main.cs        All (9,1)      (C#/l Abbrev)----1-
```

```
#include <stdio.h>

int main(int argc, char *argv[]) {
  printf("Hello, World!");
  return(0);
}
-u:--  hello.c        All (6,1)      (C/l Abbrev)----1:
```

```
GNU Emacs 22.0.91.1 (i486-pc-linux-gnu, GTK+ Version 2.10.6) of 2006-12-20 on rothera, modified by Debia
```

## Emacs Editor - Exercise

- Edit file using vi editor
- Lab 1
  - Exercise 1 (1.4)
  - Question 2
  - Page 27

## 1.5

## GNU Compiler Collection (GCC)

## GCC / G++ : Available Tools

- Compilers

| | | |
|---|---|---|
| – C | : | *gcc* |
| – C++ | : | *g++* |

- Debuggers

| | | |
|---|---|---|
| – Text | : | *gdb* |
| – GUI | : | *emacs* |

- Integrated Development Environments (IDE)
  - *Anjuta*
  - *KDevelop*
  - *source-navigator*

# GCC / G++

- GNU is a free software / operating system.
- The plan for the GNU operating system was announced in September 1983 by Richard Stallman and software development work began in January 1984.
- The project to develop GNU is known as the GNU Project, and programs released under the auspices of the GNU Project are called GNU packages or GNU programs.

## GCC / G++

- Licensing
  - In order to ensure that GNU software remains free, the project released the first version of the GNU General Public License (GNU GPL) in 1989.
  - The GNU Lesser General Public License (LGPL) is a modified version of the GPL, intended for some software libraries.
  - It gives all recipients of a program the right to run, copy, modify and distribute it, while forbidding them from imposing further restrictions on any copies they distribute.

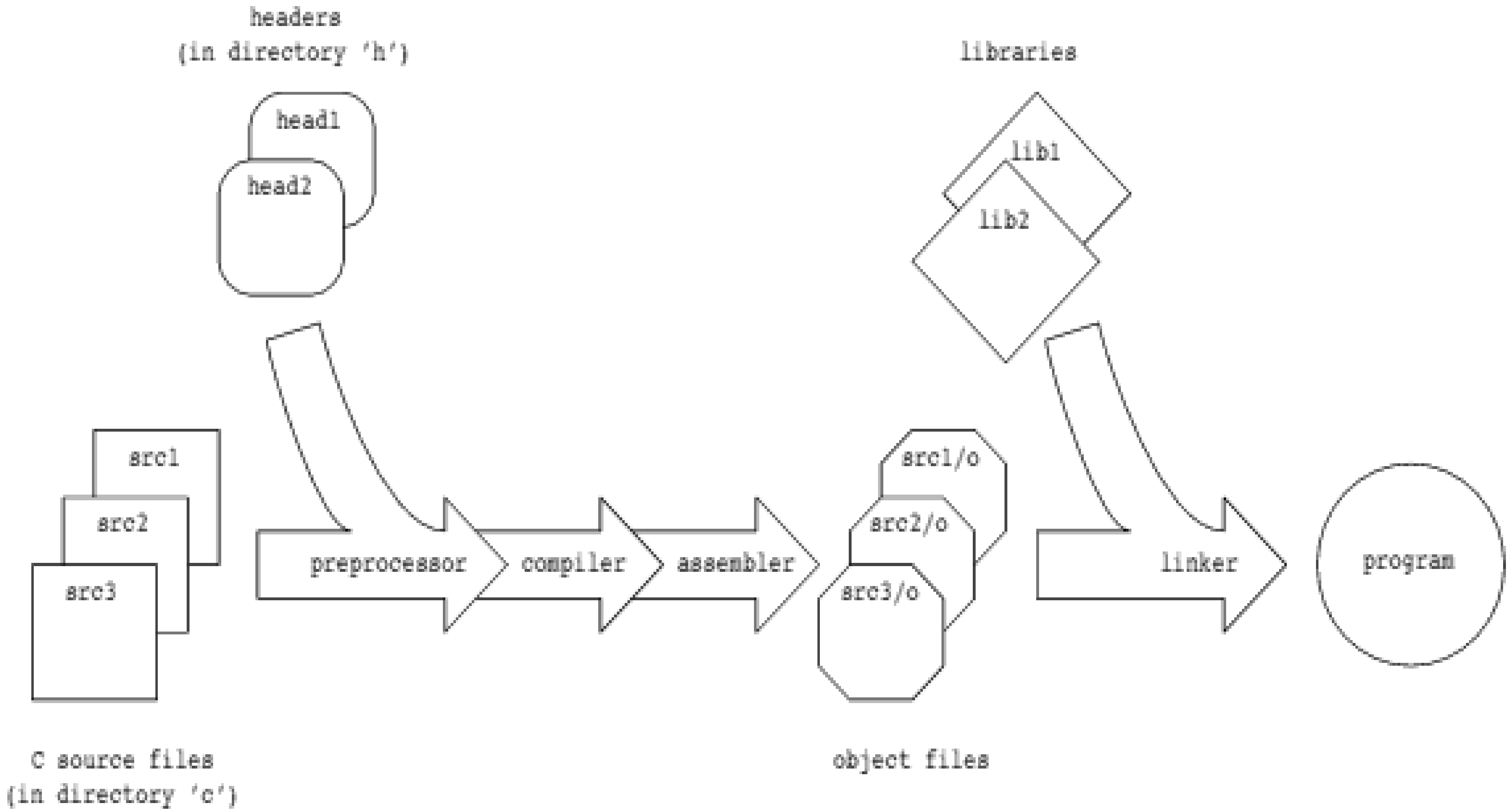
## GCC / G++

- The GNU Compiler Collection (GCC) is a set of programming language compilers produced by the GNU Project.
- It is free software distributed by the Free Software Foundation (FSF) under the GNU GPL and GNU LGPL, and is a key component of the GNU tool chain.
- It is the standard compiler for the free software Unix-like operating systems, and certain proprietary operating systems derived there from such as Mac OS X.
- Originally named the GNU C Compiler, because it only handled the C programming language, GCC was later extended to compile C++, Java, Fortran, Ada, and others.

# GCC / G++

- Standard compiler on:
  - GNU/Linux
  - Other free operating systems
  - OS X
- Used on almost every other platforms:
  - Windows
  - Embedded systems
  - UNIX and UNIX-like systems (Linux / BSD)
- Support for a very wide variety of architectures.
  - Same behavior on all systems.
  - Support for kernel/embedded programming.
- Good code generation.
- Standards compliance.

# GCC / G++ : Four Stages of Compilation



# GCC / G++ : Four Stages of Compilation

| Stage | Input | Output |
|---|---|---|
| *preprocessor* | source code | pre-processed source code |
| *compiler* | pre-processed source code | assembly source code |
| *assembler* | assembly source code | object file |
| *linker* | object files | executable file |

# GCC / G++ : Flag

- The compiler can be stopped at any stage using the appropriate flag:
  - **-E** -- stops after the preprocessing stage. It outputs source code after preprocessing to standard out (the terminal).
  - **-S** -- stops after the compile stage. It outputs the assembly for each source file to a file of the same name but with a .s extension.
  - **-c** -- stops after the assemble stage. It outputs an object file for each source file with the same name but with an ".o" extension.

# GCC / G++ : Compilation Flags

- There are also many other useful compiler flags that can be supplied. We will cover some of the more important ones here but others can be found in manual g++.
  - **-g :** includes debug symbols

    This flag must be specified to debug programs using gdb.

  - **-Wall** : show all compiler warnings.

    This flag is useful for finding problems that are not necessarily compile errors. It tells the compiler to print warnings issued during compile which are normally hidden.

  - **-o** : the name for the compiled output.

  - **-v** :Give details of what gcc is doing. Use this to help track down problems.

  - -ansi : Force ANSI compliant compilation

# GCC / G++ : Example- A basic compile

- *g++* syntax:

  *g++ [<options>] <input files>*

- Some useful options:

| | |
|---|---|
| *-g* | produce debugging info (for gdb) |
| *-Wall* | "Warnings *all*" |
| *-ansi* | force ANSI compliant compilation |
| *-D<sym>* | define *<sym>* in all source files |
| *-o <name>* | output filename |

- Example:

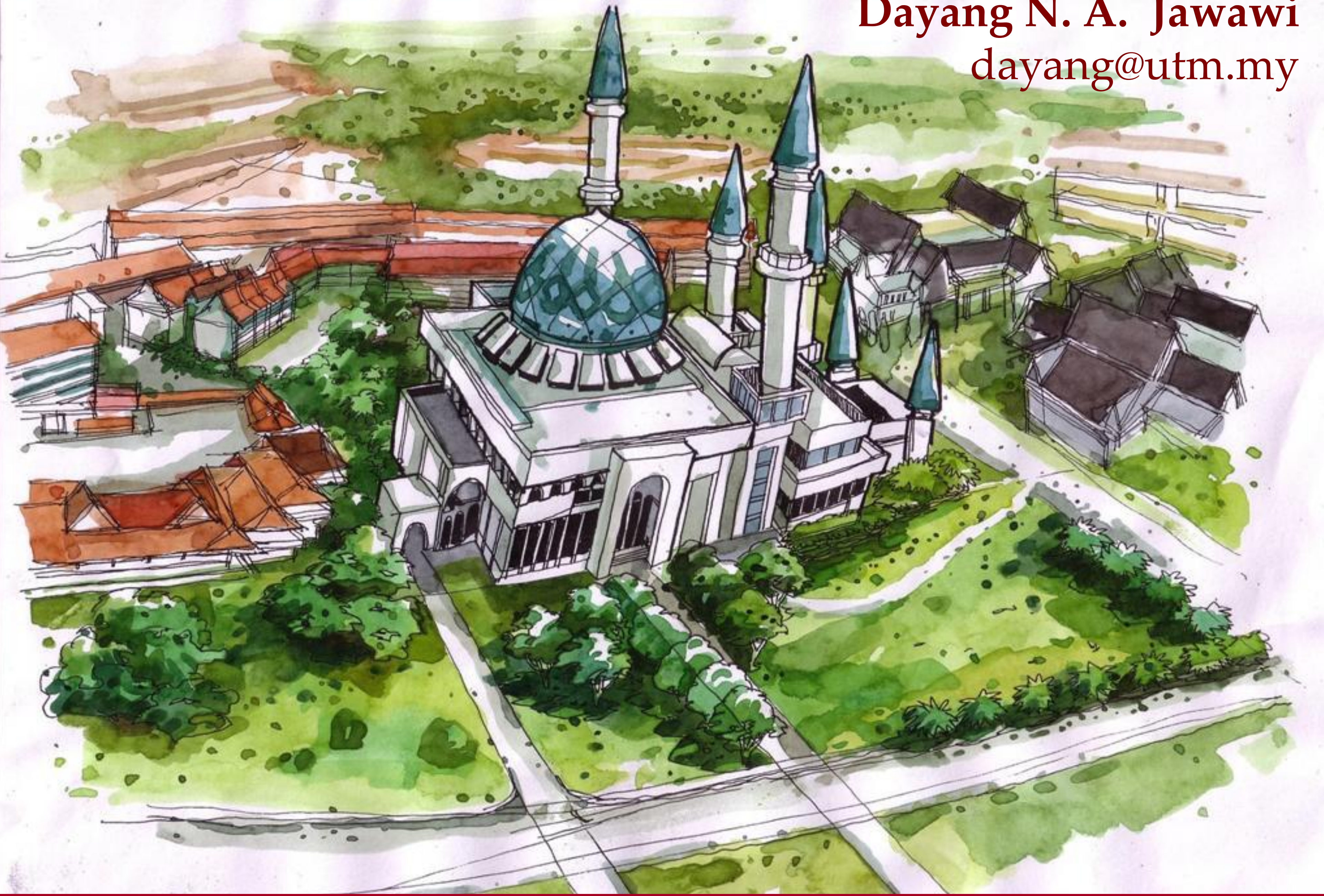
  *g++ -g –Wall –ansi –o hello hello.cpp*

## G++ Compile - Exercise

- Compile and run the programs you wrote using vi and emacs
- Lab 1
  - Exercise 2 (1.5)
  - Question 1 & 2
  - Page 37-39

## Revision - Exercise

- Solve the problems given
- Lab 1
  - Exercise 3 (1.3)
  - Question 1(file), 2 (array) & 3 (structure)
  - Page 45-47
- Solve the problem in pair, a student solve Question 1 and the other one student solve Question 2.
- Extend the both solutions to use structure, as the example specified in Question 3. Use the same concept for Question 1.
- Exchange your solutions and create a table to check your friend’s solution.
- Submit online elearning by Monday 2nd March 2015 the program (.cpp file) and the table your friend created to check your solution.



**Dayang N. A. Jawawi**
dayang@utm.my

**http://comp.utm.my/dayang/**