Introduction to C, C++, and ~~Unix~~ Unix/Linux

CS 60

UC Santa Barbara

Lecture 1: Introduction

T. Gonzalez (original by M. Turk)

Today's agenda

- Course overview
- Get to know each other
- {Write, compile, run} our first C program

Introduction to C, C++, and Unix/Linux

- Catalog description:

  Syntax and semantics of C and C++. Introduction to basic UNIX utilities and tools. Students complete several small projects that exercise their understanding of the material presented in class.

  Prerequisite: CS 20 (and CS 10 and Math 3B).

  Would be a good idea to have had CS 30 or ECE 15AB, but it is not required.

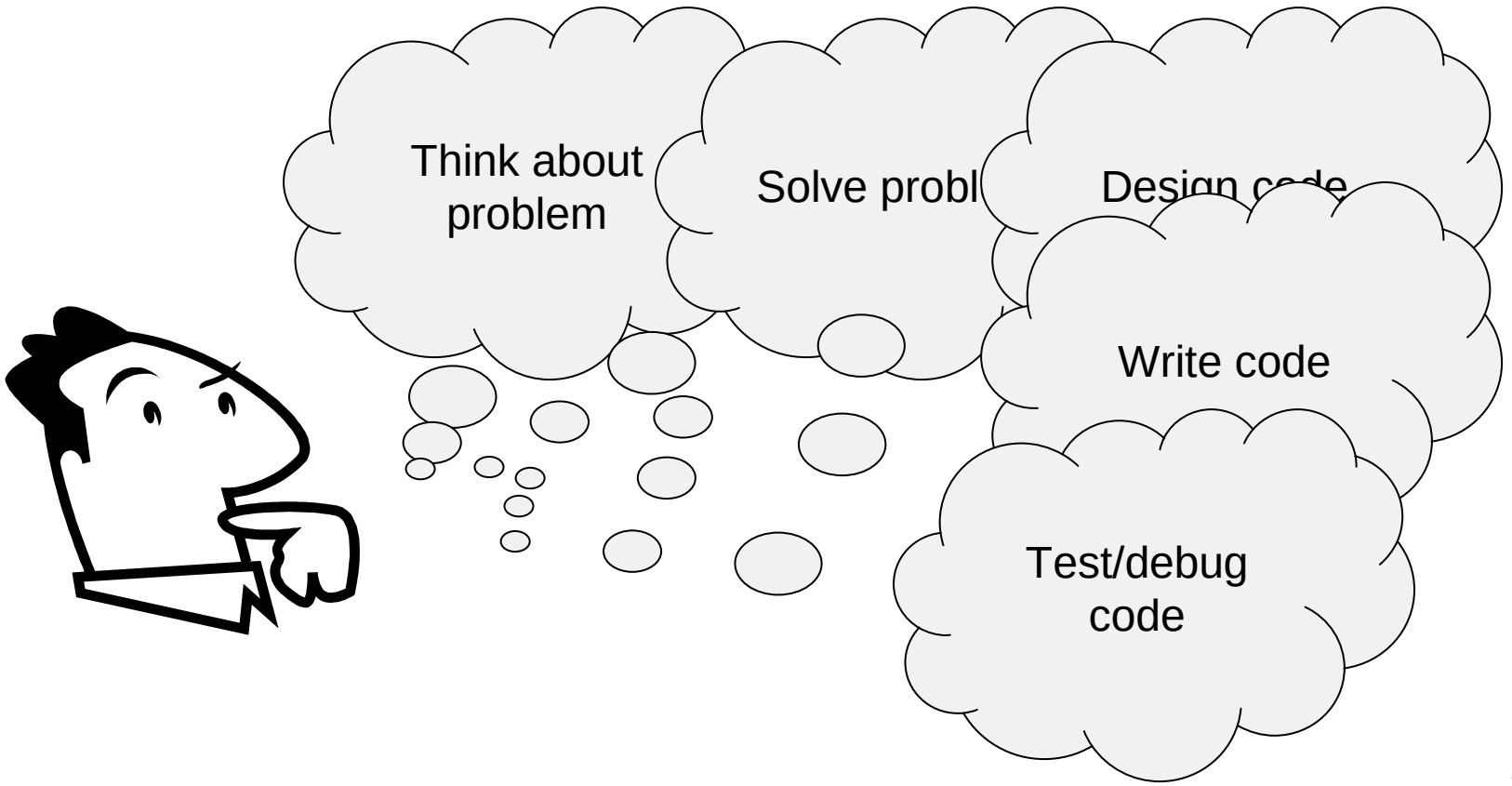
In this course, we will gain good understanding …

- Learn the C language
- Learn to use the Unix/Linux/GNU software development environment
- Learn the C++ language
- Write a lot of programs – learn by doing

- Let's begin with all of this ….

Programming is…

- Implementing an idea/algorithm
  - Mapping a solution to the particular system and language constraints



Constraints

- Computer system
  - Mainframe, server, PC, Xbox, PDA, cell phone, automobile…
  - Windows, MacOS, Unix, Linux, Symbian…
  - CPU speed, memory, I/O…
- Languages
  - Assembly, Basic, Fortran, Pascal, C, C++, Java, Lisp, C#,…
  - Scripting: C shell, perl, javascript…
- APIs and libraries API – Application Program(ming) Interface
  - C standard library, math library, runtime library, OpenGL…
- Integrated Development Environments (IDEs)
  - Borland C++, GNU, Visual Studio, NetBeans…

Writing good code

- Need to first understand the algorithm well
  - Or else you'll really be in trouble!
- Consider carefully how to map the algorithm to a computer program
  - Have to understand data types, data structures, control flow, file I/O, etc.
- Good code design
  - Readable, modular, extensible, reusable by others
  - Documentation: lots of useful comments
  - Meaningful variable names, logical (member) functions
  - Make it easy to debug
  - Good intuitive user interface
  - Efficient, easy to modify, etc., etc….

Course details

Slides are in:

http://www.cs.ucsb.edu/~teo/cs60.m07/map.html

Almost everything else is in

http://www.cs.ucsb.edu/~teo/cs60.html

- Syllabus
- Schedule
- Links
- Announcements (via e-mail)
- Assignments (given out in class)

Assignment #0

- Due Wednesday (2:00 PM)
  - Get a CSIL account or CSIL access (if you have an Engr account)
  - Join the cs60 course discussion list
  - Send an email to cs60@cs that, in your own words, very briefly describes your understanding of the policy on academic integrity for this course and states clearly whether or not you intend to fully abide by it. PLUS add to the e-mail a list of all the programming courses you have taken as well as other programming experience.

For next class

- Reading
  - Kernighan and Ritchie, **The C Programming Language**
    - Chapter 1
  - Siever, **Linux in a Nutshell**
    - Chapters 1 and 2

- We **will** have a Thursday quiz this week!
  - Bring pencil/pen

What we expect from you

- A lot – this is not an easy course, it is tough
  - Though it is reasonable and we will be fair
- Be responsible for everything that takes place in class (lectures and discussion sessions)
- Check the class web site, map.html web page, and your e-mail regularly
- Keep up:
  - Do the reading on time
  - Do the assignments on time

What we expect from you (cont.)

- Ask questions – to classmates, to TA, and to me
  - But don't wait until the last minute!
- Feel free to disagree, ask questions, and ask for clarifications
- Be here on time for Thursdays quizzes
- Turn in your programs on time

Simplest C program

```
main()
{ }
```

Next simplest C program

```
main()
{
  int x;
  x = 1 + 2 + 3 + 4;
}
```

Another simple C program

```
main()
{
  double x, pi;

  pi = 3.1415927;
  x = 45.0*pi/180.0;
}
```

Yet another simple C program

```
main()
{
  int i, x;
  for (i=0; i<100; i++)
     x += i;
}
```

For all of these…

```
$ gcc file.c
```

compiles the source code into an executable file called “a.out” (the default)

```
$ gcc -o prog file.c
```

produces an executable file called “prog”