C Programming Lecture Series

16th August

IIT Kanpur

About ‘the’ Course

- An assignment based course
- More emphasis on problem solving

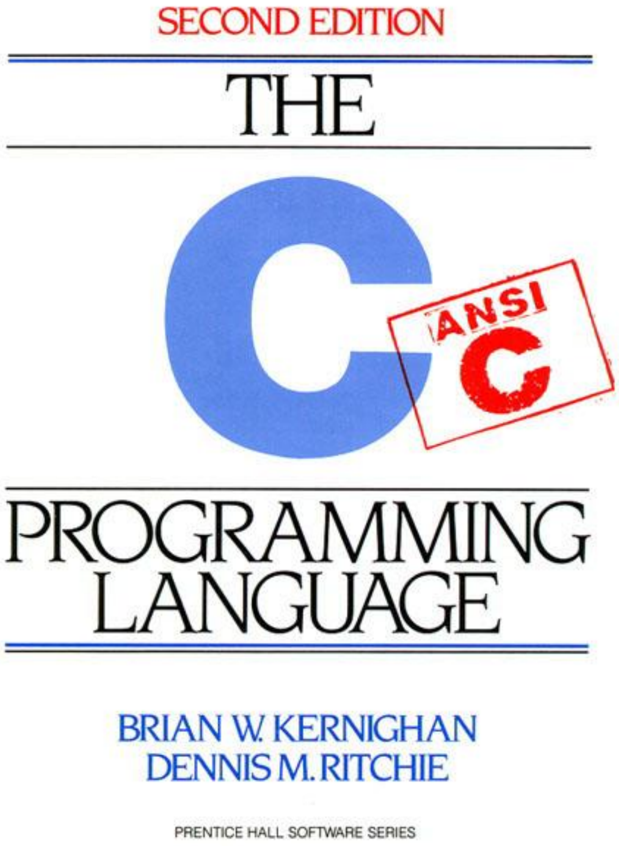
Instructors:

Deepak Majeti.	mdeepak@
Nitin Munjal.	nitinm@
Rishi Kumar.	rishik@
Satendra Kumar Yadav.	satendra@

Topics to be covered

- Introduction to Programming in C & Restricted Exposure to Linux - Today
- Data, Operators, I/O - Tomorrow
- Conditional Expressions, Control Flow – 23rd Aug.
- Loops
- Functions for structure and Recursion
- Pointer and Arrays
- Dynamic allocation
- Structures and Applications, Storage Classes
- Pre-processor, File Handling, Math library
- Algorithms: searching, sorting

Text



SECOND EDITION

THE

C

ANSI

C

PROGRAMMING LANGUAGE

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES

Kernighan, Ritchie. Second Edition

Course website

- Website (slides, important updates)

http://students.iitk.ac.in/programmingclub/course/

- Discussion page (lecture clash, doubts)

http://students.iitk.ac.in/programmingclub/course/discuss.html

About C

- *GNU : GNU's Not Unix*
  - *GNU C: gcc is a standard compiler*
- *C is non portable*
  - *Terms: Compiler (human -> machine [once]), Interpreter (instructions -> machine [each time the program is run])*
- *C is a high level language*
  - *One line in c maps to many lines of assembly code*

My first C program!

```
/* thou shalt begin from somewhere*/
#include <stdio.h>

// program prints hello world
int main() {
   printf ("Hello world!\n");
   return 0;
}
```

More..

```
#include <stdio.h>
// program reads and prints the same thing

int main() {
   int number;
   scanf(“%d”, &number);
   printf (“%d\n”, number);
   return 0;
}
```

1. Programming on Linux

- Linux command line: GNU-C
  - Use console based editors: vi, emacs, nano
  - Or text based editors: kwrite, gedit, kate
- IDE
  - Eclipse * http://www.eclipse.org/cdt/downloads.php

* = available on windows too.

Linux Familiarization

- Common shell commands
  - Remember, commands are issued to a shell
  - pwd, ls, dir, mkdir, cd, date, whoami
  - touch, cp, mv, rm, chmod, cat, less, more, tail
  - man
  - Commands are programs (usually in /usr/bin, /bin/)
  - Most commands take options and input
    - ls   ls -a   ls -l   ls -lt   ls -ltr
- Everything is case-sensitive
- Tab completion, command history

Files, directories and permissions

- Directory
  ```
  drwxr-xr-x 2 nitinm cse 4096 2008-08-13 22:46 Pictures
  ```
- File
  ```
  -rw-r--r-- 1 nitinm cse 3446 2008-08-14 15:16 test.c
  ```
- Special files (advanced)
  - .a : static library
  - .so : shared object (dynamic)
  - Pipes : fifo / buffered   prwx--x--x
  - Device files : /dev/cdrom etc.

Programming on Linux contd…

- Writing programs
  - Use any editor (graphical, console)
  - Save file as <filename>.c
- Compiling programs
  - gcc <filename>.c   gcc funnysingh.c –o funnysingh
- Running programs
  - ./a.out   ./funnysingh
    (executable files need to have executable permissions.
    $chmod +x <executable>)

Compilation is not a single stage

- Pre process : cpp (C Preprocessor) gcc –E
  - Removes comments, includes #include files
- Compile : gcc –c (GNU compiler)
  - main step, compilation, change into machine code
- Link : ld (GNU linker)
  - link executables

gcc does all the above steps

2. C on windows

- Use a text editor
  - install notepad++
  - compiler : MinGW
    how to install and work-
    http://csjava.occ.cccd.edu/~gilberts/mingw/
- IDE
  - Eclipse *
  - Microsoft Visual C++ Express Edition 2008

Or 3. Work on windows, yet use gcc

- Install SSH Secure Shell or Putty
  - Connect to cc servers: webhome.cc.iitk.ac.in or linserv.cc.iitk.ac.in etc.
- Want to see GUI too?
  - Install Xming
    - And then, enable X11 tunnelling

- Why doesn’t my windows binary run on linux?
  - File format: exe and elf
    - man elf
  - In linux, program does system calls.
  - Libraries are different

Good programming practices

Indentation

```
#include <stdio.h>
int main() {
   printf("Hello World!\n");
   return 0;
}
```

```
#include <stdio.h>
int main() {
printf("Hello World!\n");
return 0;
}
```

Good programming practices contd..

- Variables names
  - Not too short, not too long
  - Always start variable names with small letters
  - On work break
    - Capitalize: myVariable, OR
    - Separate: my_variable

Good programming practices contd...

- Put comments

```
#include <stdio.h>
int main() {
   /* this program adds
   two numbers */
   int a = 4; //first number
   int b = 5; //second number
   int res = 0; //result
   res = a + b;
}
```

Good programming practices

- Your code may be used by somebody else
- The code may be long
- Should be easy to understand for you and for others
- Saves lot of errors and makes debugging easier
- Speeds up program development