Tutorial & Laboratory

Programming & Data Structure: CS11001/19001

Section - 4/D

DO NOT POWER ON THE MACHINE

Department of Computer Science and Engineering

I.I.T. Kharagpur

Spring Semester: 2013 - 2014 (**09.01.2014**)

Instructors

Name	Tel. No. & email
Debashis Mukherjee	**9433664720** & debashis_mukherjee@yahoo···
Jimmy Jose	**78724 17861** & jimmy@cse···
Sandipan Sikdar	**9143247344** & sikdarsandipan99@···
Saurav Kumar Ghosh	**9674410464** & saurav.kumar.ghosh@···
Shreyasi Das	**9433638807** & shreyasidas.kolkata@···
Shyantani Maiti	**9433774182** & shyantani.maiti@···
Mj. Vikram Singh	**9126143871** & vikramsingh.eme@···

cse.iitkgp.ernet.in, gmail.com

Tutorial

1. Carry a C programming book and a notebook (tutorial) for the tutorial and laboratory classes.
2. Solve tutorial problems in the notebook with proper dates. Almost everyday the notebook will be asked for evaluation.

Feedback

The number of students

- who have done computer programming:
- who have done programming in C, C++, Java or Pascal:
- who have worked on Unix or Linux OS:

Laboratory Classes

9th Jan., 16th Jan., 23nd Jan., 30th Jan, 6th Feb., 13th Feb., 27th Feb., 6th Mar., 13th Mar., 20th Mar., 27th Mar., 3rd Apr., 10th Apr., 17th Apr. 2013

Dates of Laboratory Test:
I: 13th February,
II:10th April.

Laboratory Marks Distribution

1. First class - **Introduction** (will not be evaluated).
2. *Assignment Evaluation (a):* **40**
3. *Laboratory Test I (l_1):* **25**
   (13th February)
4. *Laboratory Test II (l_2):* **35**
   (10th April)
5. *Marks Obtained (t):* $\mathbf{a + l_1 + l_2}$.

Normalization

Marks will be normalized over seven sections of Programming and Data Structure Laboratory.

The Computing Environment

Thin-Client and Server

The Server

- Sun Fire X4170 Server
  - Intel Xeon 5570, 4-Core, 8-Thread,
  - 64-bit AMD64 (x86-64) architecture,
  - Clock 2.93 GHz,
  - *Cache*:

    L1: 32 KB (I) + 32 KB (D) per core,

    L2: 256 KB per core,

    L3: 8 MB per processor,

- *Main Memory* : **144 GB** (Max),
- *Hard Disk* : 2× 146 GB.

OS, Editor and Compiler

- *OS* : **SunOS**
- *Programming Language* : **C**
- *Compiler* : **gcc** (**cc**)
- *Editor* : **gedit**
- *Command-interpreter*: **bash**

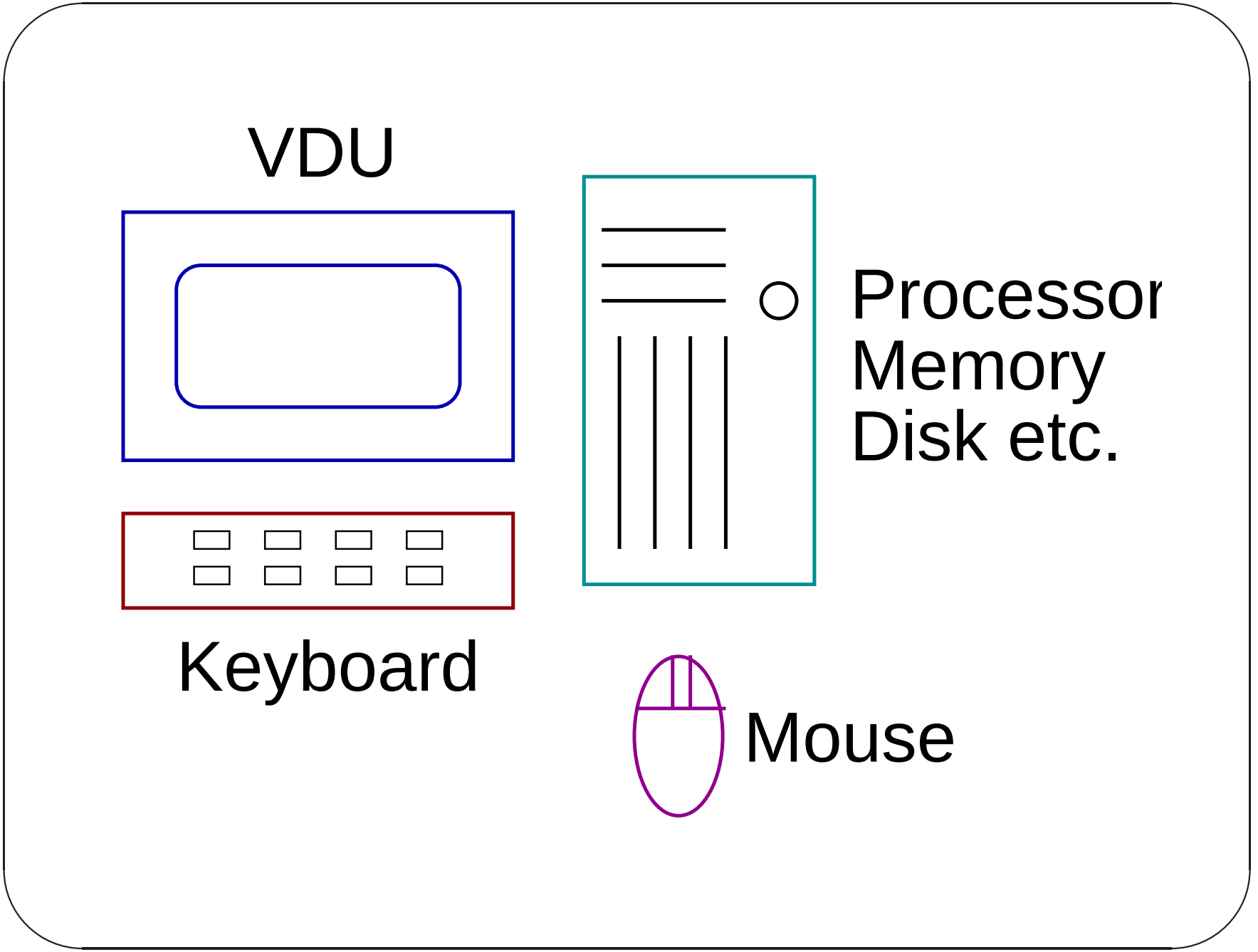
Thin Client

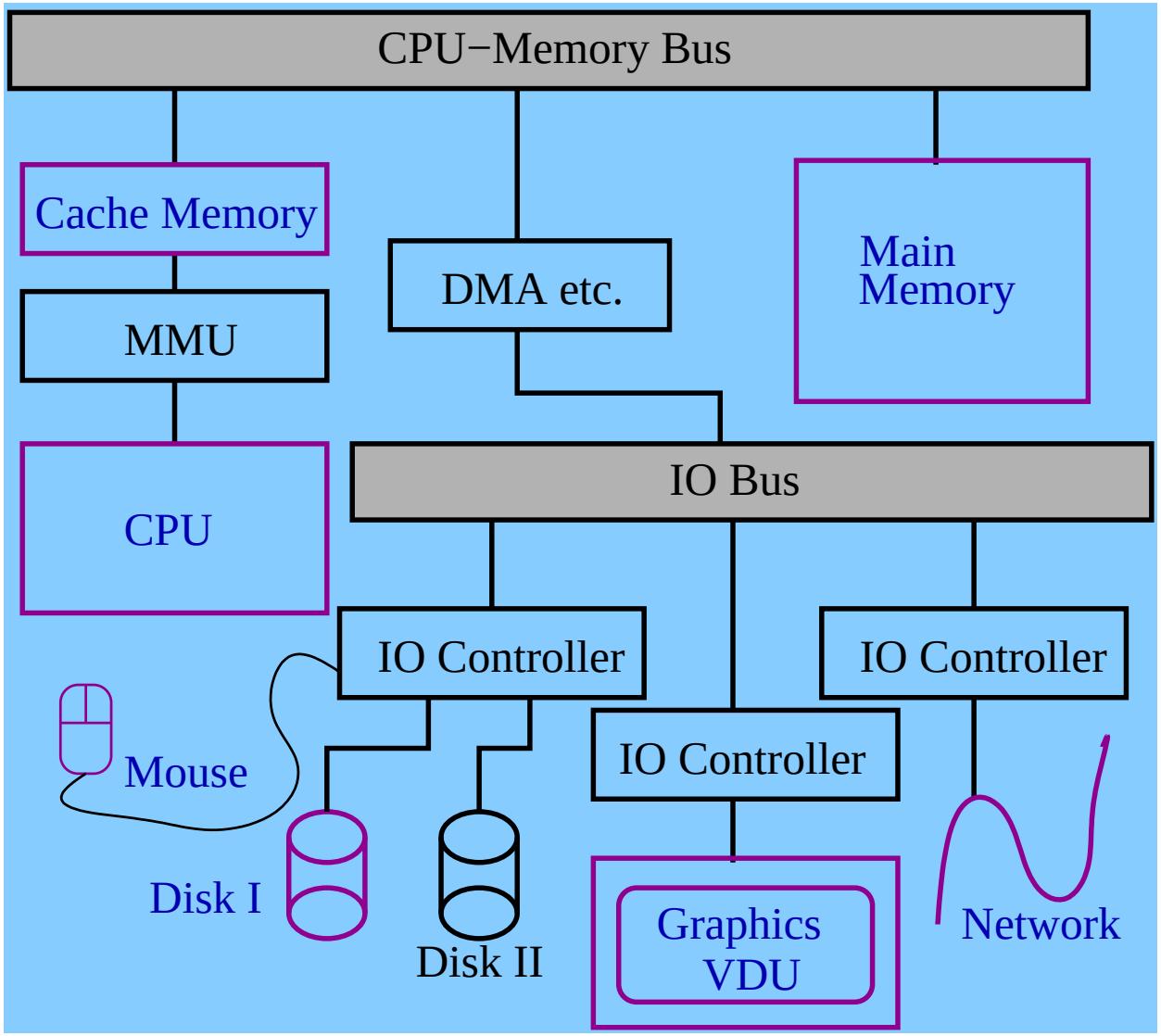
A thin client is a computer (and necessary software) that runs most its computational job on a more powerful server. Large number of clients share the same server.
A modern thin client is a computer terminal that provides graphical interface to the user. But for other functionality it uses the server.

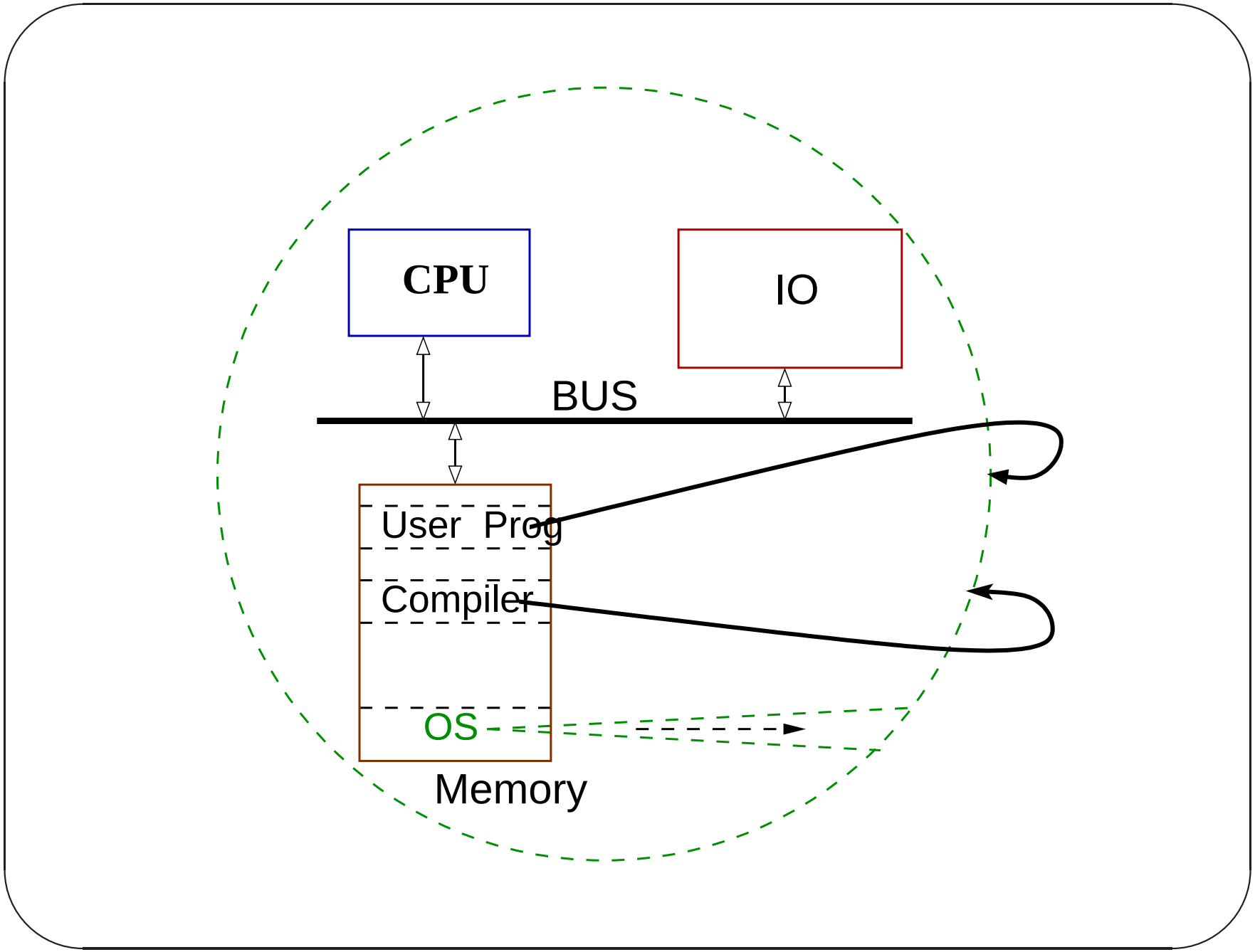
Thin Client

- Sun Ray Display Client -
- *Display*: 17-inch, LCD, flat-panel,
- *Graphics*: Integrated 24bit, 2D graphics,
- *I/O*: USB mouse, keyboard, smart-card reader (ISO-7816-1), audio (CD-quality audio in/out), microphone,
- *Network*: 10/100 BASE-T connection to LAN,
- *Ports*: USB, serial, audio, video.

A Computer


VDU
Processor
Memory
Disk etc.
Keyboard
Mouse


CPU−Memory Bus
Cache Memory
MMU
CPU
DMA etc.
Main Memory
IO Bus
IO Controller
IO Controller
IO Controller
Mouse
Disk I
Disk II
Graphics VDU
Network


CPU
IO
BUS
User Prog
Compiler
OS
Memory

Central Processing Unit (*CPU*)

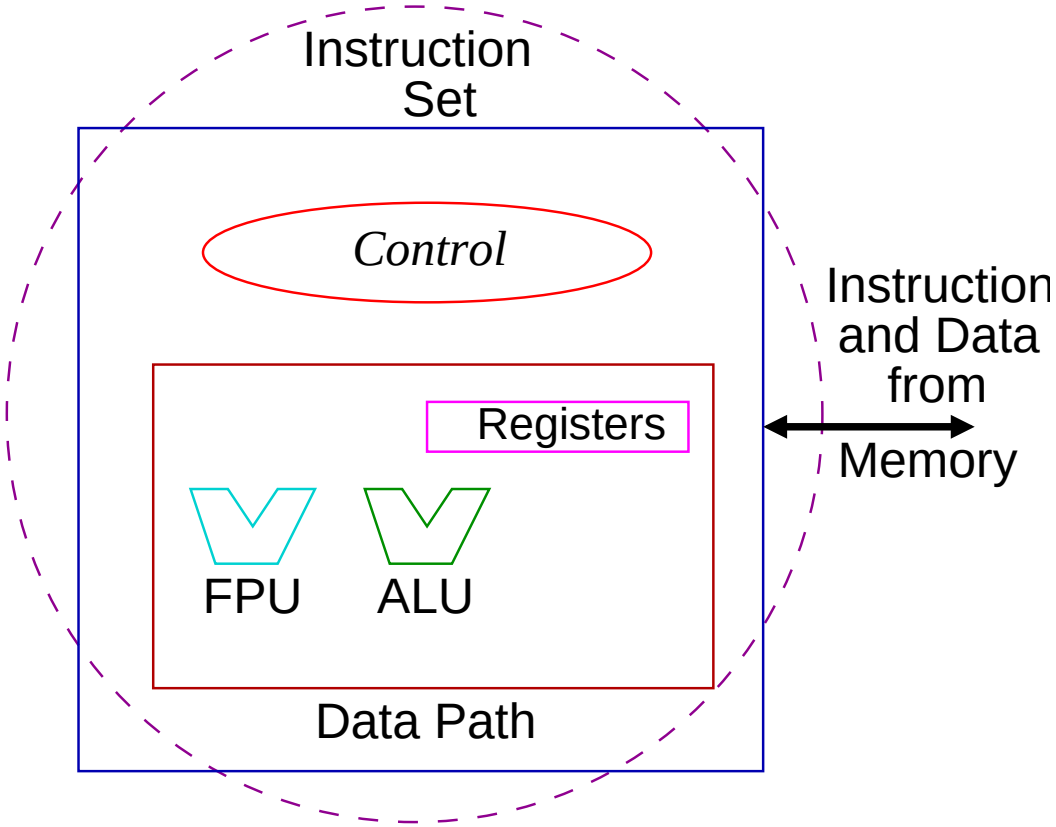


Figure 1: **A CPU**

Memory Subsystem

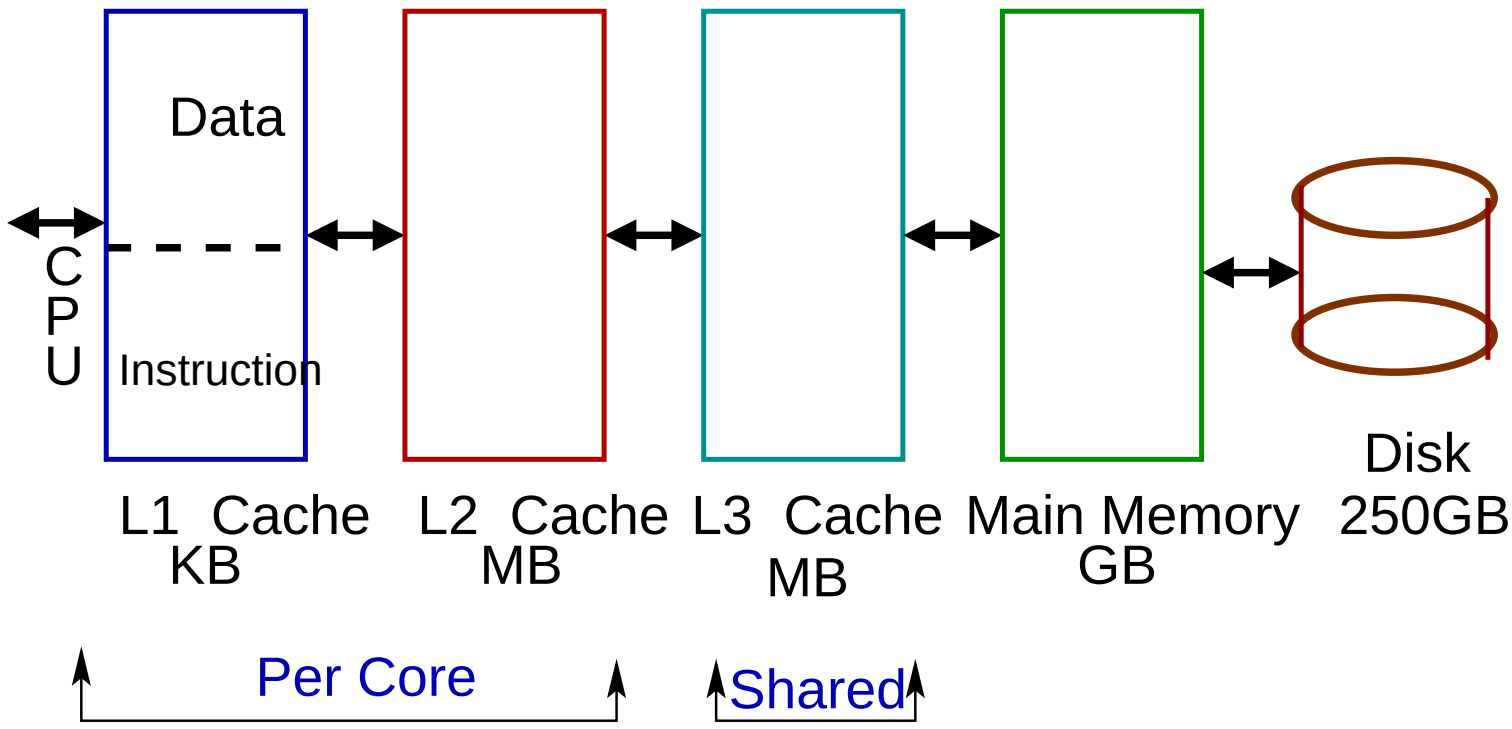
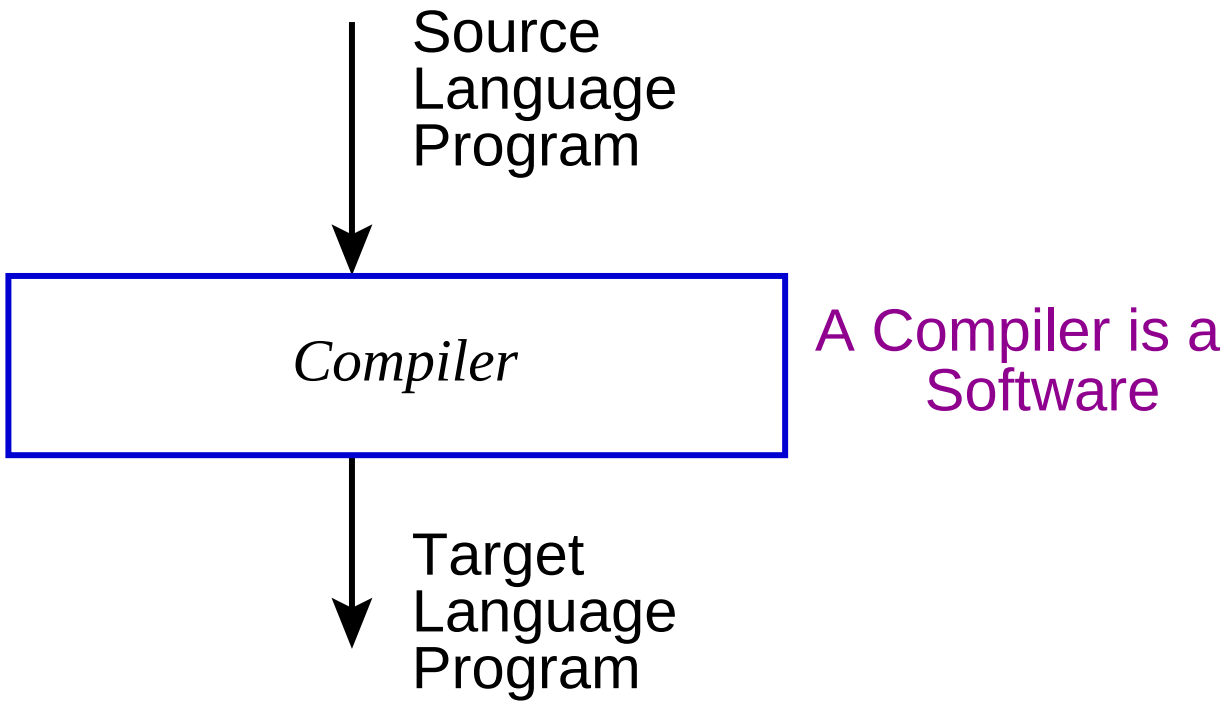


Figure 2: **Computer Memory**

A Compiler



A Compiler is a Software

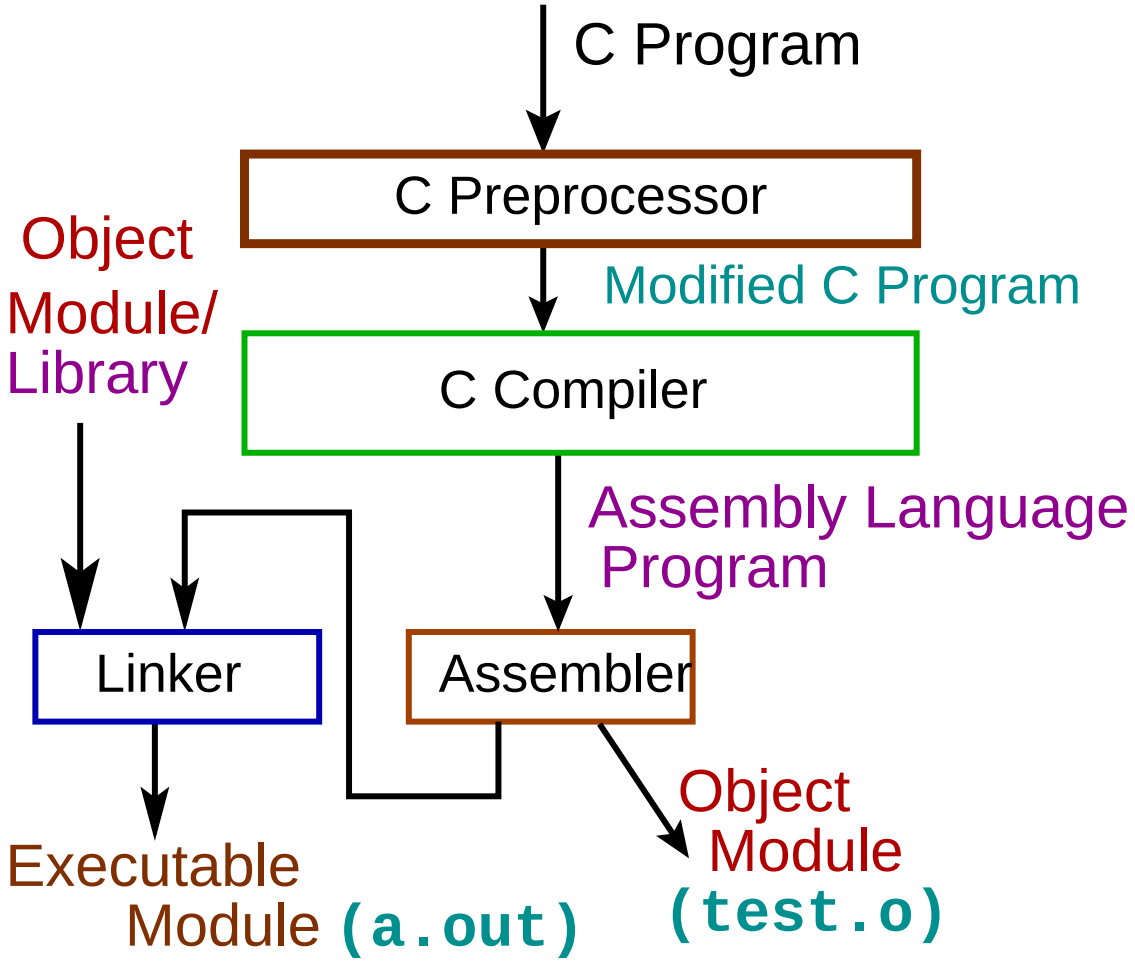


Figure 3: **Preprocessor + Compiler + Assembler + Linker**

C Program: `aCprog.c`

```
/*
 *  C Program: aCprog.c
 */
#include <stdio.h>
int main()
{
    printf("This is a C Program\n");
    return 0 ;
}
```

How to Compile a C Program?

```
$ gcc -Wall aCprog.c
```

This command[a] produces an executable file (module) a.out. A compiler can also produce other types of output.

[a] The command may be `cc` in place of `gcc` in your machine.

Compile & Execute

```
$ gcc -Wall aCprog.c
$ ./a.out
This is a C Program
$ ls -l aCprog.c a.out
-rw-rw-r-- 1 goutam goutam 118 Dec 31
10:43 aCprog.c
-rwxrwxr-x 1 goutam goutam 4752 Dec 31
10:43 a.out
```

Basic Steps

- How to Start a Computer?
- How to write a C Program?
- How to Compile a C Program to an Executable Module?
- How to Run (Create a Process with) an executable Module?

How to Start your Session and how to Finish it?

1. Enter the user name, d*nn*, (*n* or *nn* is the machine number e.g. d7 or d17); then enter the password, pds123.
2. Open one or two terminals on the VDU using the right-button of the mouse.

1. One terminal is active at a time - use the mouse to activate a terminal.
2. bash can be terminated and a terminal may be closed by pressing Ctrl-D.

Log Out the Session

1. Close all windows.
2. logout: use Launch menu to Log Out.

How to Start the gedit Editor

- Select an window and enter the following command

  ```
  bash-3.00$ gedit &
  ```

- A gedit window is created. This is the place where you write your program.
- Use the Save manue to save the file. It will ask for file name for a new file.
- For an existing file us Open menue.

How to Start the gedit Editor

- Close the gedit window selecting the File → Quit menue.
- You may select the font size using Edit → Preferences.
- We shall discuss about other features of gedit in due course.

Write, Compile and Execute a C Program

1. Start the gedit editor: gedit &
2. Select File → New. You get the scratchpad/buffer to write your program.
3. Save the program with a file name e.g. first.c
4. Write and save the program using Save menue.

```c
/*
 * This is the first Program: first.c
 */
#include <stdio.h>
int main()
{
    printf("Tamaso Ma Jyotir Gamaya\n") ;
    return 0 ;
}
```

1. Run the compiler with your program as data:

   $ gcc -Wall first.c

2. If the compilation process ends without any message, the file 'first.c' contains a well-formed C Program and it has been translated to the executable module '`a.out`'

1. Run the executable module:

   $ ./a.out

2. It will flash the message

   Tamaso Ma Jyotir Gamaya

   on the VDU screen.

If There is an Error

```
/*
 * This is the first Program: first.c
 */
#include <stdio.h>
int main(
{
    printf("Tamaso Ma Jyotir Gamaya\n") ;
    return 0 ;
}
```

Compiler Message

```
$ gcc -Wall first.c
first.c:6:  error:  syntax error before
'{' token
```

What to DO?

1. Open the existing file first.c in gedit.
2. Identify and fix the errors.
3. Save the modified file (same name).
4. Compile it again.

Any time you make a change in the file, you have to save it and compile it to get the new executable module, a.out.

1. If there is no more message from the compiler, the C program is syntactically correct (well-formed).
2. An well-formed C program may have logical error. Error-free compilation does not guarantee the logical correctness of a program.

Getting the Slides

- Run Mozilla and access

  `http://cse.iitkgp.ac.in/~goutam`

- Follow the link

  `Programming & Data Structures ...` of Spring: 2013 - 2014.

- Download the correct .ps/.pdf file.

A Few Linux Shell (*bash*) Commands

- `ls` - Lists the directory contents.
- `rm` - Removes files from a directory.
- `pwd` - Prints name of the current/working directory.
- `mkdir` - Makes Directories.
- `rmdir` - Removes empty directories.

A Few Linux Shell (*bash*) Commands

- `cd` - Changes working directory.
- `cp` - Copy files and directories.
- `mv` - Moves/Renames files.
- `man` - Format and display the online manual pages.

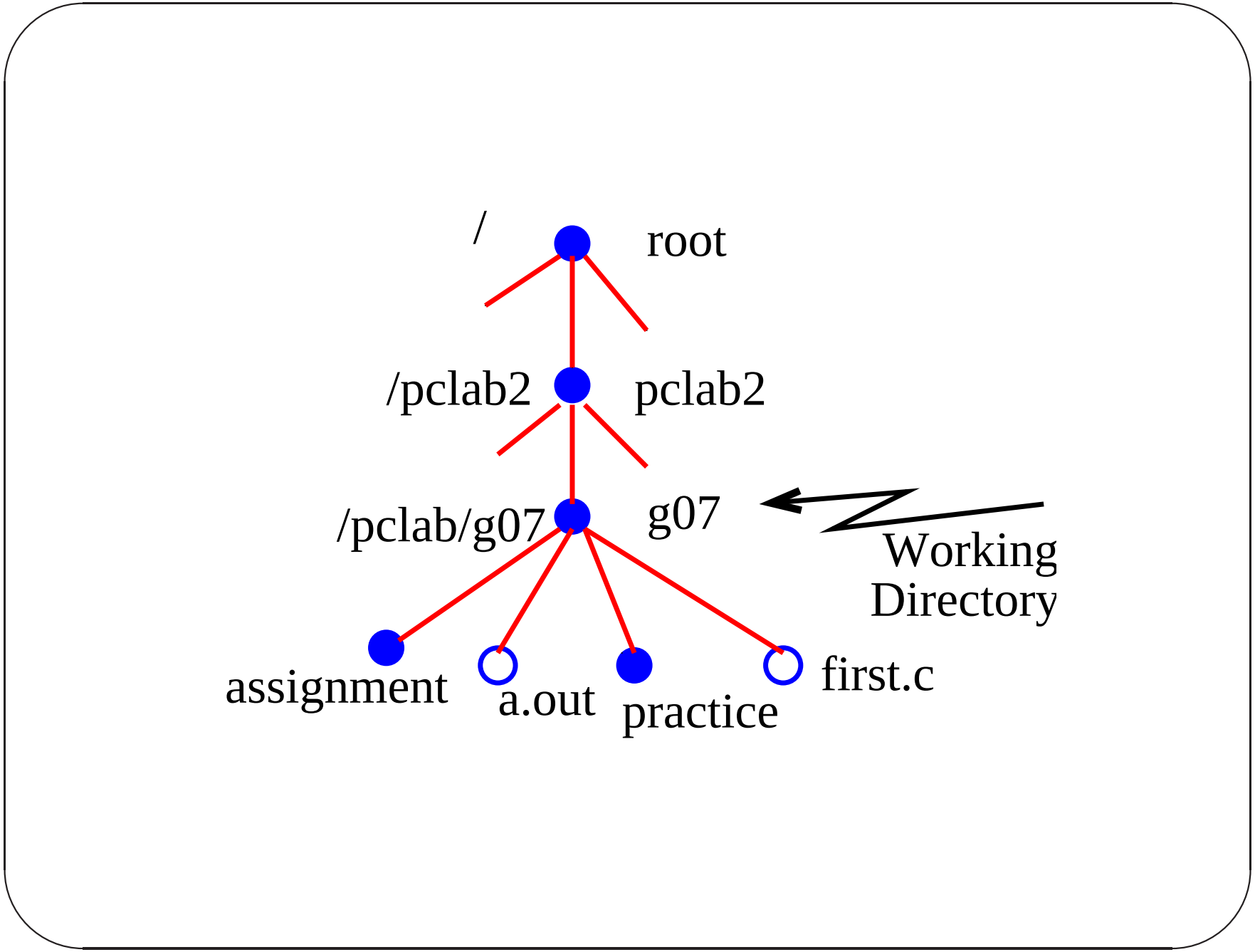
File System

- program and data are stored in a non-volatile media as named objects called files.
- A file may contain - C program, object file, executable file, data, etc.
- Files are organised in a systematic way.

File System

- There are special files (in Unix/Linux) called directories containing names of files and subdirectories.

Directory Structure on a Machine


/
root
/pclab2
pclab2
/pclab/g07
g07
Working
Directory
assignment
a.out
practice
first.c

Creating Directories

1. `Create` a subdirectory practice01 under your working directory.
2. `Move` the first.c under practice01.
3. `Change` current directory to practice01.

Write, Compile and Execute

Example - II

```
/* second.c */
#include <stdio.h>
int main() {
    int n ;

    printf("Enter a non Negative integer: ") ;
    scanf("%d", &n) ; /* Reads an integer */
    printf("\nSum of (0 + ... + %d) = %d\n",
                    n, n*(n + 1)/2 ) ;
    return 0 ;
}
```

Example - III

```
/* Area of a Circle: third.c */
#include <stdio.h>
#include <math.h>
int main() {
    float radius, area ;

    printf("Enter the radius :") ; scanf("%f", &radius) ;
    area = 4.0 * atan(1.0) * radius * radius ;
    printf("\n Circle-Area = %f for Radius = %f\n",
                              area, radius) ; return 0 ;
}
```

Compile with Mathematical Library

It may be necessary to compile with mathematical library.

```
$ cc -Wall -lm third.c
Where is the mathematical library?
$ cd /lib
$ ls -l libm*.*
-rwxr-xr-x 1 root root 177315 Dec 20
2004 libm-2.3.4.so
lrwxrwxrwx 1 root root 13 Jul 12 2006
libm.so.6 -> libm-2.3.4.so
```

Example - IV

```
// Convert Fahrenheit to Celsius: fourth.c
#include <stdio.h>
int main(){
    float cel, fah ;
    printf("Enter temp. in F: ") ;
    scanf("%f", &fah) ;
    cel = 5.0*(fah-32.0)/9.0 ;
    printf("%6.2f F = %6.2f C\n", fah, cel) ;
    return 0 ;
}
```

Example - V

```
#include <stdio.h>
int main() {    // **** fifth.c
    int first, second, max ;

    printf("Enter two integers :") ;
    scanf("%d%d", &first, &second) ;
    if(first > second) max = first ;
    else max = second ;
    printf("\nMax(%d,%d) = %d\n",
           first, second, max) ; return 0 ;
}
```

Example - VI

```
#include <stdio.h>
int main() {          // **** sixth.c
    int n, i, sum = 0 ;

    printf("Enter a non-negative integer:") ;
    scanf("%d", &n) ;
    for(i = 0; i <= n; ++i) sum += i ;
    printf("\nSum of (0 + ... + %d) = %d\n",
                                  n, sum ) ;
    return 0 ;
}
```

Example - VII

```
int sum(int n) {  // *** seventh.c
    if(n==0) return 0;
    else return n+sum(n-1);
}
#include <stdio.h>
int main() {
    int n ;
    printf("Enter a non-negative integer:") ;
    scanf("%d", &n) ;
    printf("\nSum of (0+...+%d)=%d\n",n,sum(n));
    return 0 ;
}
```

Try to Modify the Given Programs

1. Modify example-II to print the sum of an AP series, where the first term (a) and the common difference (b) and number of terms (n) are three inputs.
2. Modify example-III to calculate the volume of a sphere where the radius is the input.
3. Modify example-V to read three integers and find the smallest.

What Does it Do?

```
#include <stdio.h>
int what(int n, int m) {
    if(m==0) return n;
    else return what(m,n%m);
}
int main() {
    int m, n ;
    printf("Enter two non-negative integers:") ;
    scanf("%d%d", &n, &m) ;
    printf("\nwhat(%d, %d) = %d\n",m,n,what(m,n));
    return 0 ;
}
```