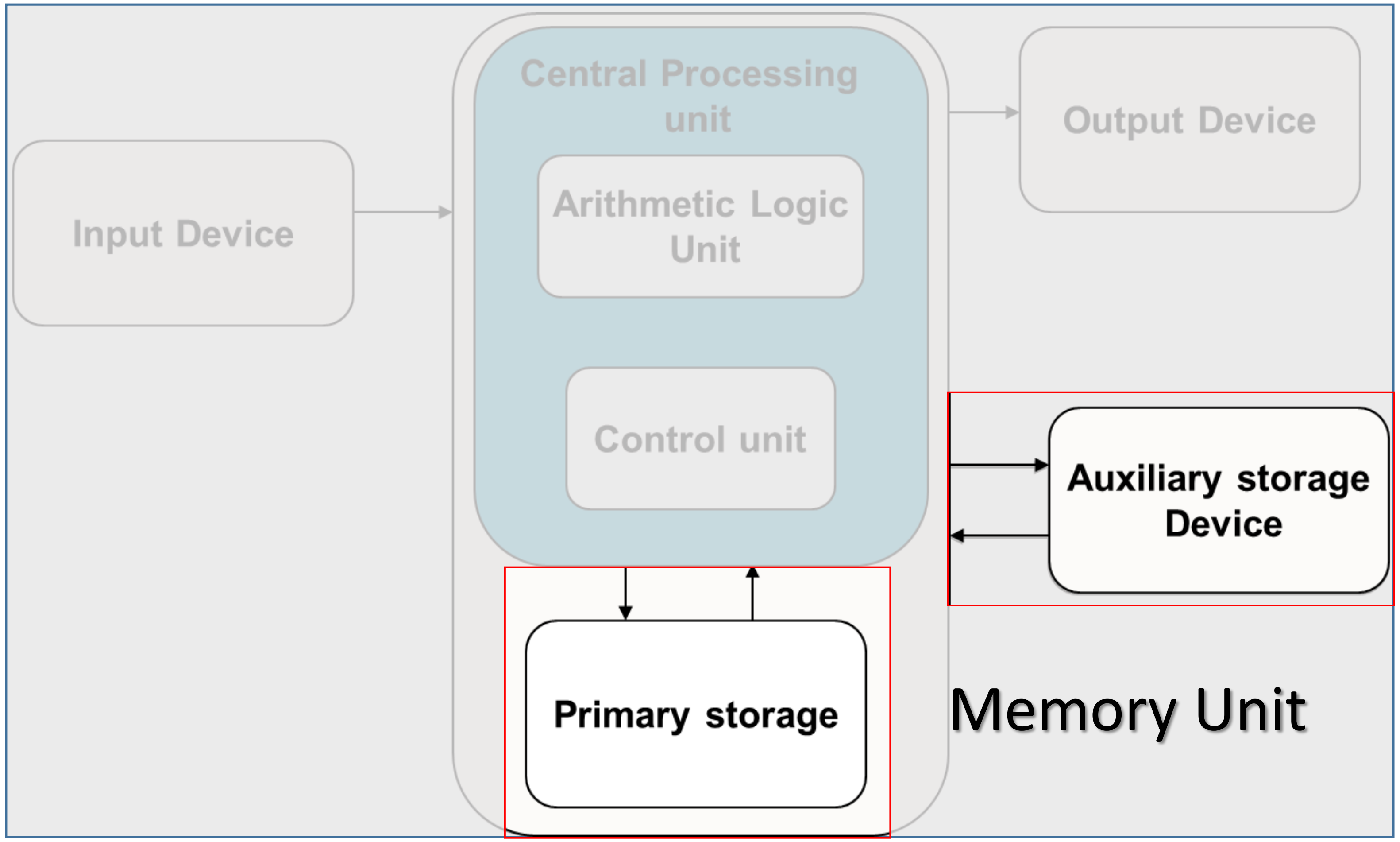
Introduction to Computing

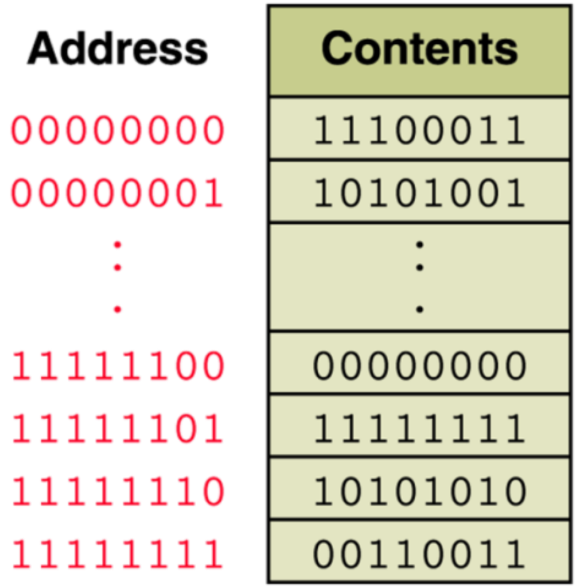
S1_2

Computer Organization



Memory unit

- ➢Storage device where the data and instructions fed by the user are **stored**
- ➢An **ordered sequence of storage cells**, each capable of holding a piece of **information**
  - ➢Each cell has its own **unique address**
- ➢The information held can be input data, computed values, or your program instructions.



Address	Contents
00000000	11100011
00000001	10101001
:	:
11111100	00000000
11111101	11111111
11111110	10101010
11111111	00110011

Memory unit

- ➢The computer memory is measured in terms of **bits, bytes** and **words.**
- ➢A **bit** is a **binary digit** either 0 or 1.
- ➢A **byte** is unit of memory and is defined as sequence of 8 bits.
- ➢The **word** can be defined as a sequence of 16/32/64 bits or 2/4/8 bytes respectively depending on the machine architecture.

Computer memory **classification**

- Main memory-Primary storage
- Secondary memory-Auxiliary storage
- Cache memory

Main memory

- Memory where the data and instructions, currently being executed are stored
  - Located outside CPU
  - High speed
  - Data and instructions stored get erased when the power goes off
- Also referred as **primary** / **temporary** memory
  - Semiconductor memory
  - Measured in terms of megabytes and gigabytes

Primary storage: RAM & ROM

- RAM stands for **Random Access Memory**
  - ➢ Read and write memory
  - ➢ Information typed by the user are stored in this memory
  - ➢ Any memory location can be accessed directly without scanning it sequentially (random access memory)
  - ➢ During power failure the information stored in it will be erased → volatile memory
- ROM stands for **Read Only Memory**
  - ➢ Permanent memory and non volatile
  - ➢ Contents in locations in ROM can not be changed
  - ➢ Stores mainly stored program and basic input output system programs

Secondary memory

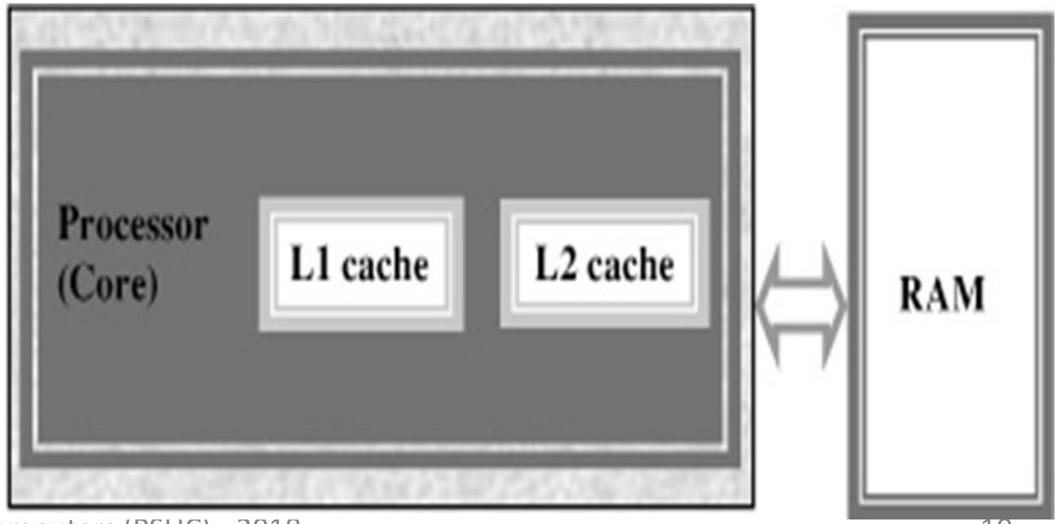
- Main memory is volatile and limited
  - Hence it is essential for other types of storage devices where programs and data can be stored when they are no longer being processed
- Installed within the computer at the factory or added later as needed

Secondary memory

- Non-volatile memory
- Made up of magnetic material
- Stores large amount of information for long time
- Low speed
- Holds programs not currently being executed

Cache memory

- High speed memory placed between CPU and main memory
- Stores data and instructions currently to be executed
- More costlier but less capacity than main memory
- Users can not access this memory



Memory System (Video)

Operating System

- ➢OS is an **integrated collection** of programs which make the computer operational and help in executing user programs.

- ➢ It acts as an **interface** between the man and machine.

- ➢ It **manages** the system resources like memory, processors, input-output devices and files.

  - ✓e.g. Windows, Linux, DOS

Computer Languages

- Machine Language- The only programming language available in earlier days
  - Consists of only **0's** and **1's**; e.g.:- 10101011
- Symbolic language or Assembly language-
  - **symbols** or **mnemonics** are used to represent instructions
  - hardware specific
  - ✓ e.g. MASM : ADD X,Y; Add the contents of y to x
- High-level languages- **English like** language using which the programmer can write programs to solve a problem.
  - more concerned with the problem specification
  - not oriented towards the details of computer
  - ✓ e.g. C, C++, C#, Fortran, BASIC, Pascal etc.

Language Translator

- **Compiler** : Program that translates entire high-level language program into machine language at a time. e.g. C, C++ compilers.

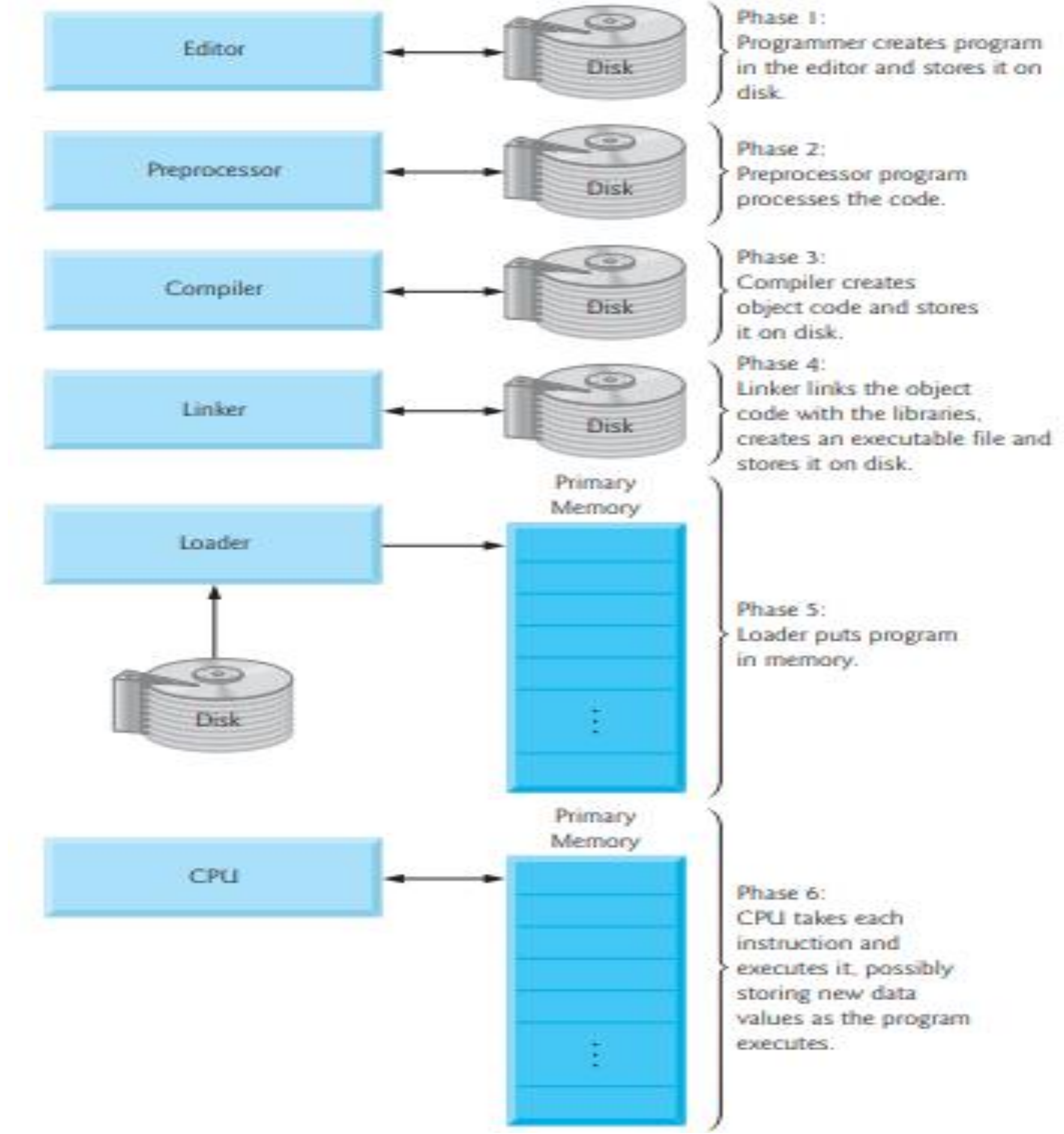
- **Interpreter** : Program which translates one statement of a high-level language program into machine language at a time and executes it.

  e.g. Basic Interpreters, Java Interpreters.

- **Assembler** : Program which translates an assembly language program into machine language.

  e.g. TASM(Turbo ASseMbler), MASM(Macro ASseMbler).

Typical C program development environment


Editor
Disk
Phase 1:
Programmer creates program in the editor and stores it on disk.
Preprocessor
Disk
Phase 2:
Preprocessor program processes the code.
Compiler
Disk
Phase 3:
Compiler creates object code and stores it on disk.
Linker
Disk
Phase 4:
Linker links the object code with the libraries, creates an executable file and stores it on disk.
Loader
Primary Memory
Disk
Phase 5:
Loader puts program in memory.
CPU
Primary Memory
Phase 6:
CPU takes each instruction and executes it, possibly storing new data values as the program executes.

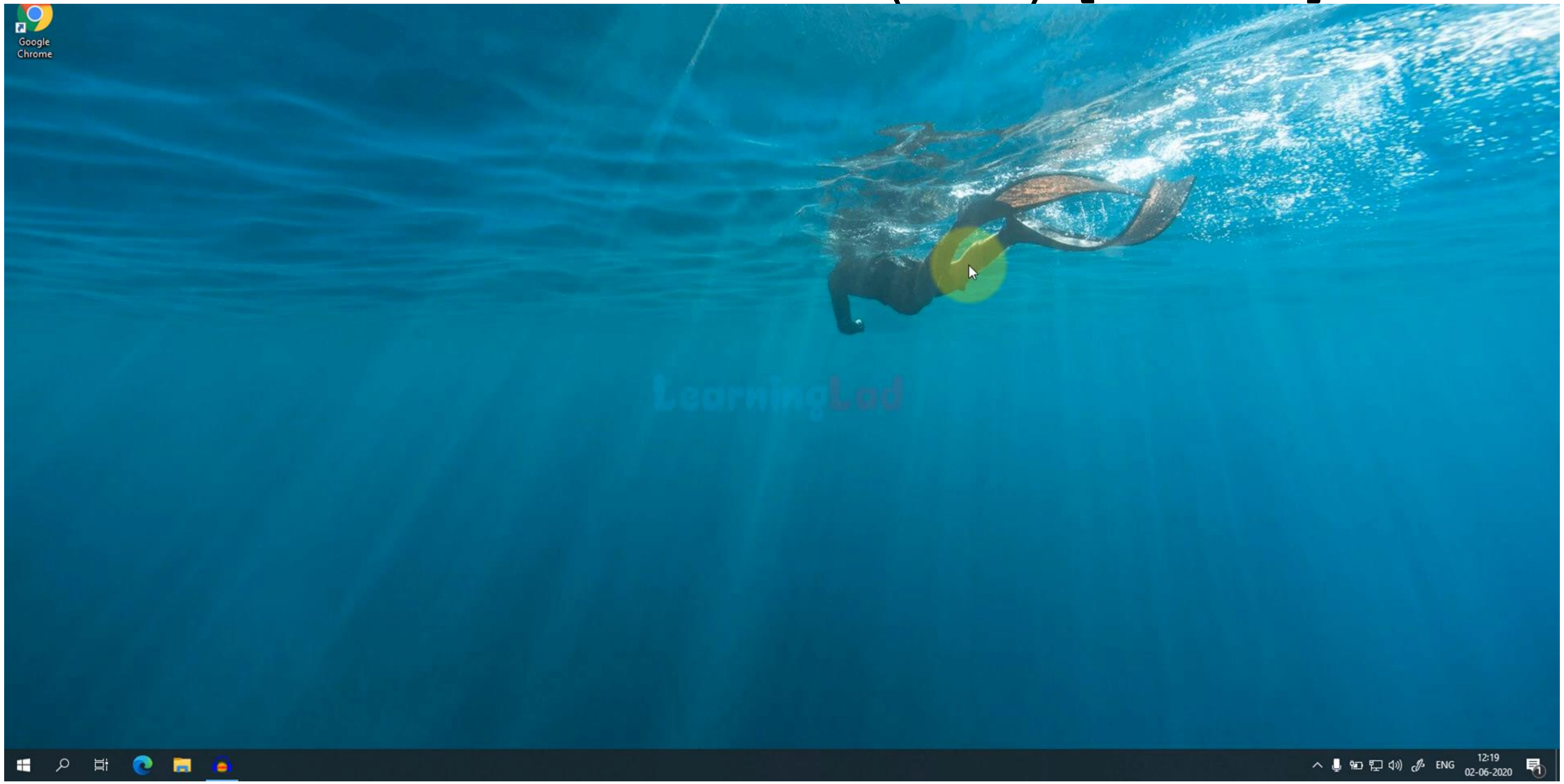
Typical **C program** development environment

C programs typically go through six phases to be executed.

These are: edit, preprocess, compile, link, load and execute

- ✓ Phase 1 : creating a program
- ✓Phases 2 and 3: Preprocessing and Compiling a C Program
- ✓Phase 4: Linking
- ✓Phase 5: Loading
- ✓Phase 6: Execution

How to Install Code Blocks (IDE) [Video]



My first **C** program

#include<stdio.h> → preprocessor directive

int main() → starting point of execution of the ‘C’ program

{ → Signifies beginning of the ‘C’ program

printf(“Hello World”);

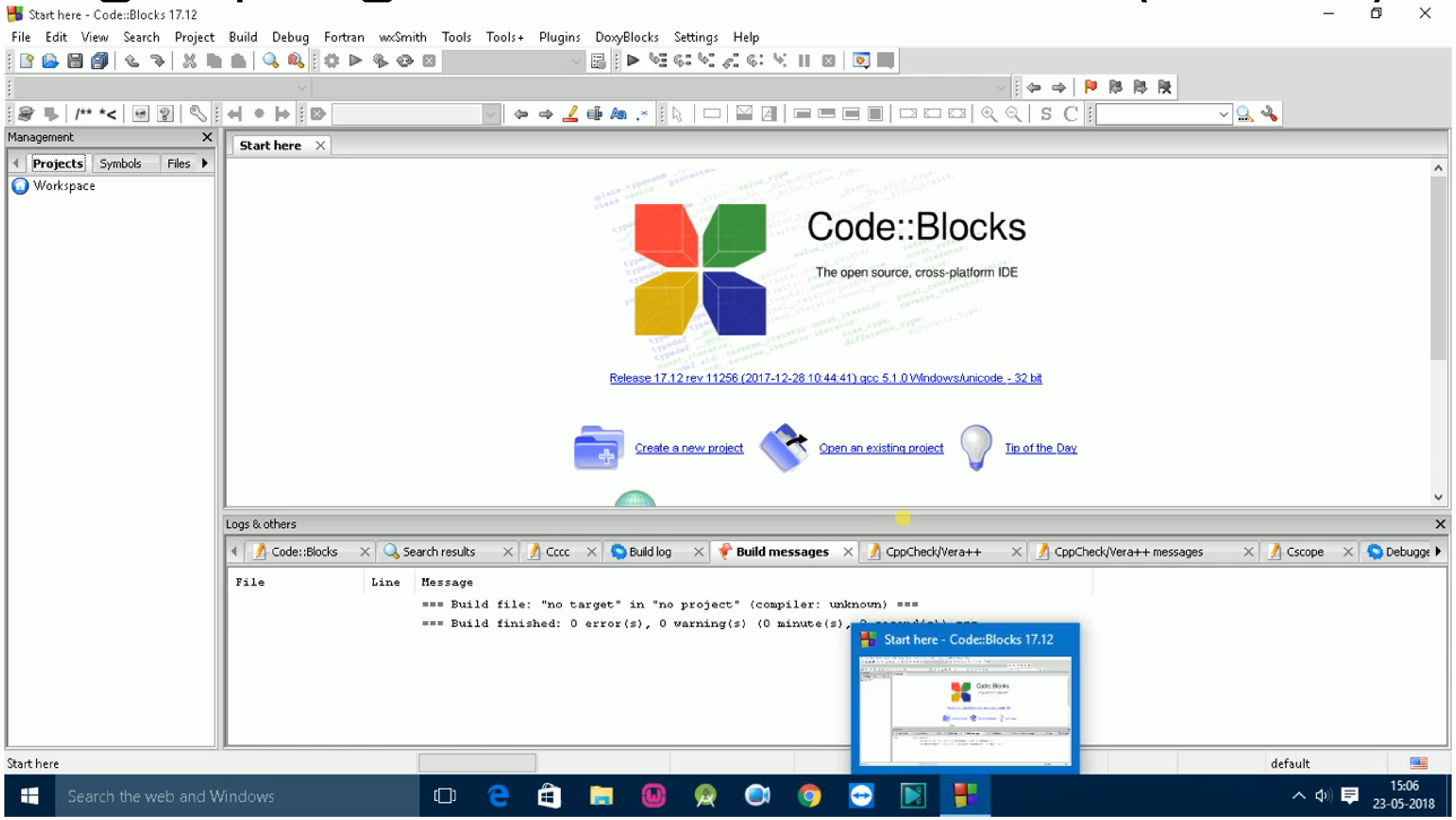
return 0;

$\}$ → Body of the ‘C’ Program

} → Signifies ending of the ‘C’ program

- ***printf***- This statement is used to output any message on the screen or console (message within the double quotes)

Running C program in Code Blocks (Video)


Start here - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
Start here
Code::Blocks
The open source, cross-platform IDE
Release 17.12 rev 11256 (2017-12-28 10:44:41) gcc 5.1.0 Windows/unicode - 32 bit
Create a new project
Open an existing project
Tip of the Day
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugge
Start here - Code::Blocks 17.12
Start here
default
Search the web and Windows
15:06
23-05-2018

```
File                 Line    Message
                             === Build file: "no target" in "no project" (compiler: unknown) ===
                             === Build finished: 0 error(s), 0 warning(s) (0 minute(s),
```

Summary

- ✓Memory System
- ✓Operating system
- ✓Different computer languages
- ✓Typical C program development environment
- ✓Code::Blocks Integrated Development Environment (IDE)

Go to posts/chat box for the link to the question **PQn. S1.2**

submit your solution in next 2 minutes

The session will resume in 3 minutes