Problem Solving Through Programming Using C

Code-CS181106

Module 1

Lima Basumatary
Asst.Professor
EE Department
BBEC, Kokrajhar

Contents

- Components of computer system.
- System software
- Operating system
- Language Processor
- Device Drive
- Application software
- Utility software

Components of computer system

- Computer requires various units to perform the prescribed tasks and to co-ordinate the operations.
- Basically a computer has four units.

1. Input unit
2. Output unit
3. Central Processing Unit(CPU)
4. Memory

Components of computer system continued..

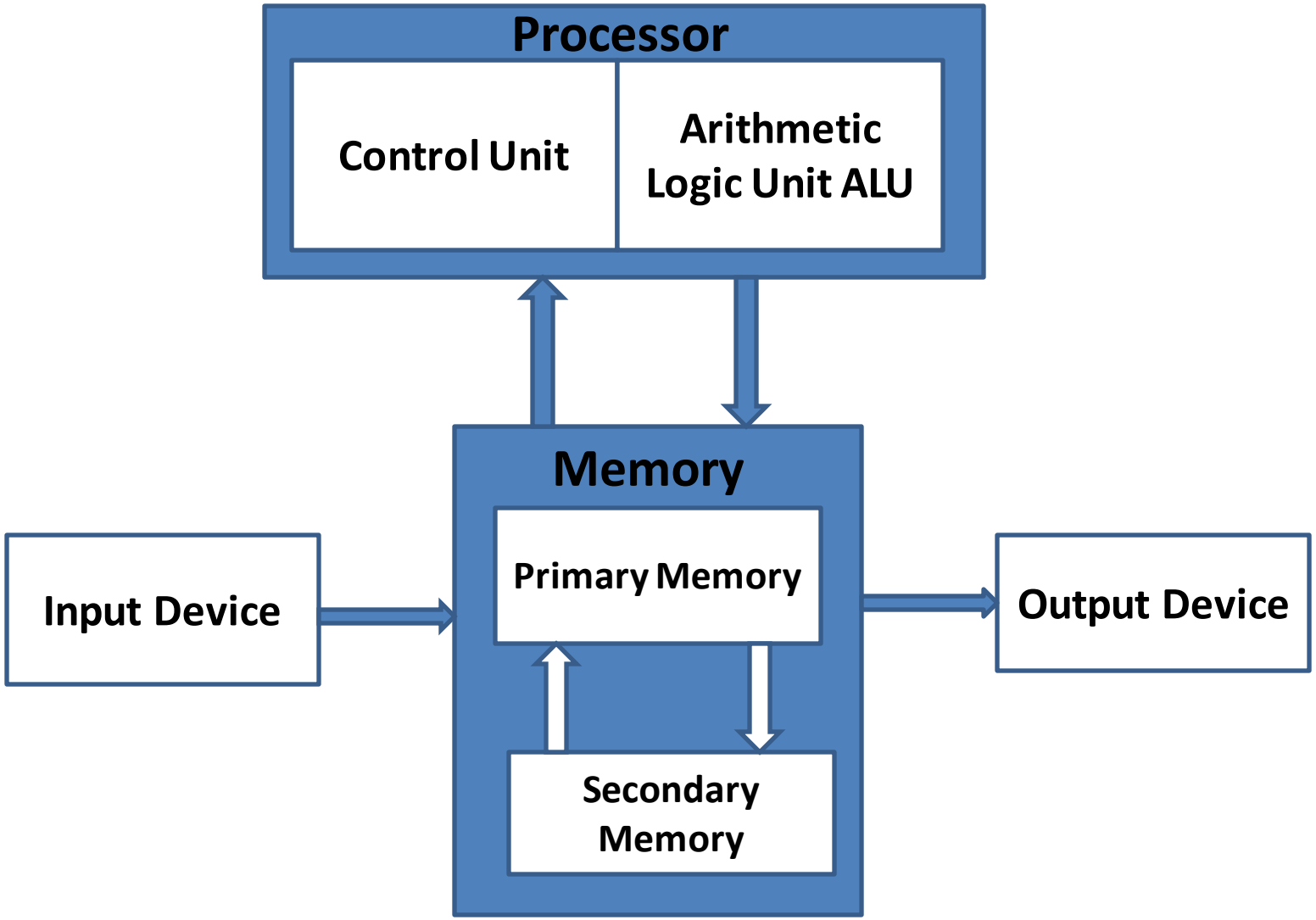


Figure 1

Components of computer system continued..

Input Unit

- It accepts the list of instructions and data from the outside world.

- It converts these instructions and data in computer acceptable format.

- It supplies the converted instructions and data to the computer system for further processing.

Components of computer system continued..

Output Unit

- It accepts the results produced by the computer which are in coded form and hence cannot be easily understood by us.

- It converts these coded results to human acceptable form.

- It supplies the converted result to the outside world.

Components of computer system continued..

CPU

The CPU consists of three main sub systems:

- Arithmetic Logic Unit (ALU).
- Control Unit (CU).
- Registers.

ALU

- Place where actual execution of the instruction takes place during the processing operations.
- All calculations are performed and all comparisons are made in the ALU.
- After the completion of processing, the final results which are stored in the storage unit are released to an O/P device.

Components of computer system continued..

Control Unit (CU)

- It coordinates with the input and output device of the computer.

- To maintain proper sequence of processing data, the control unit uses clock impulse.

- The basic form of CU is to fetch the instruction stored in the main memory, identify the operations and the devices involved in it and accordingly generate control signals.

Components of computer system continued..

Registers

- Register is one of a small set of data holding places that are part of a computer processors.

- Registers store data, instructions, address and intermediate results of processing.

Software

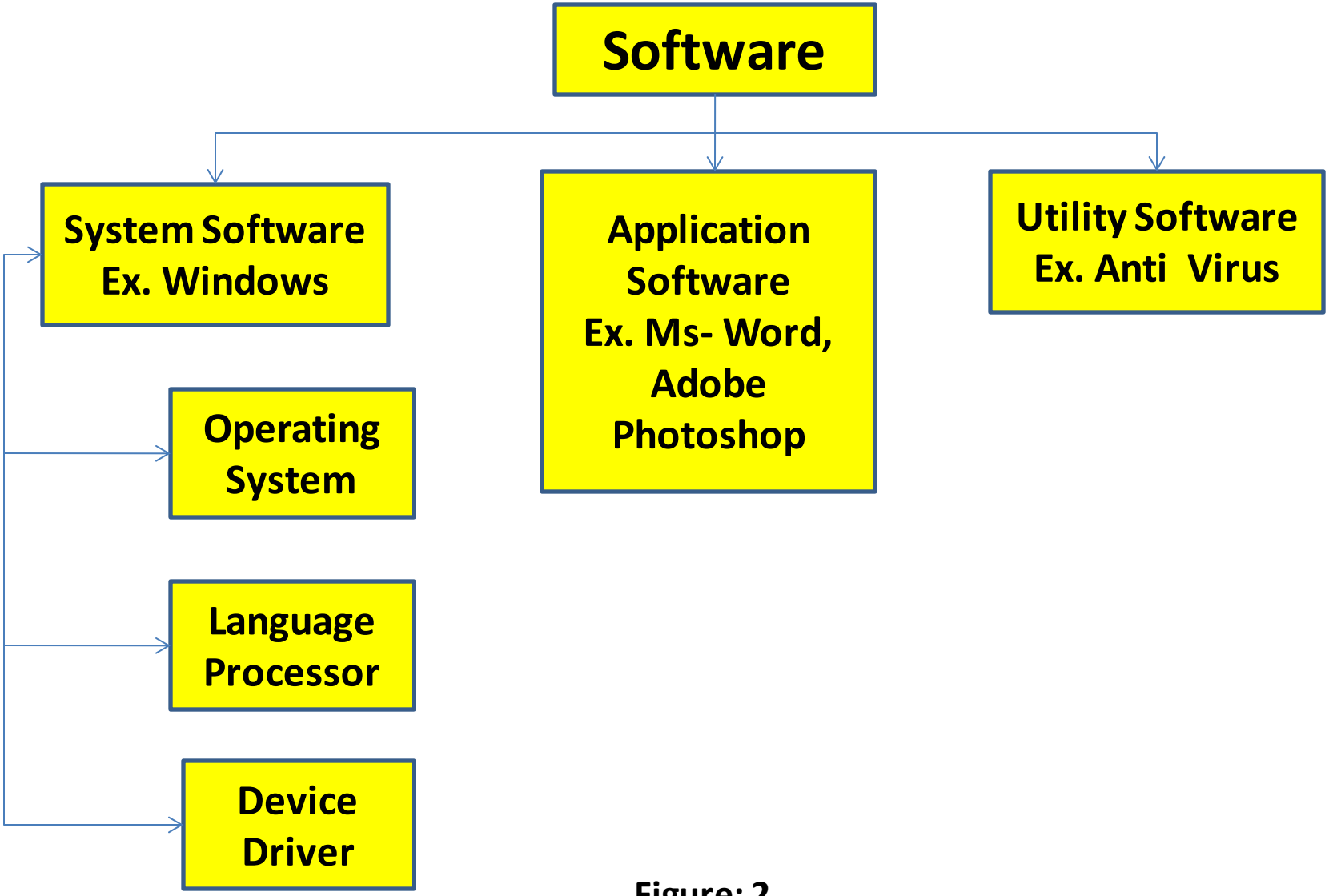


Figure: 2

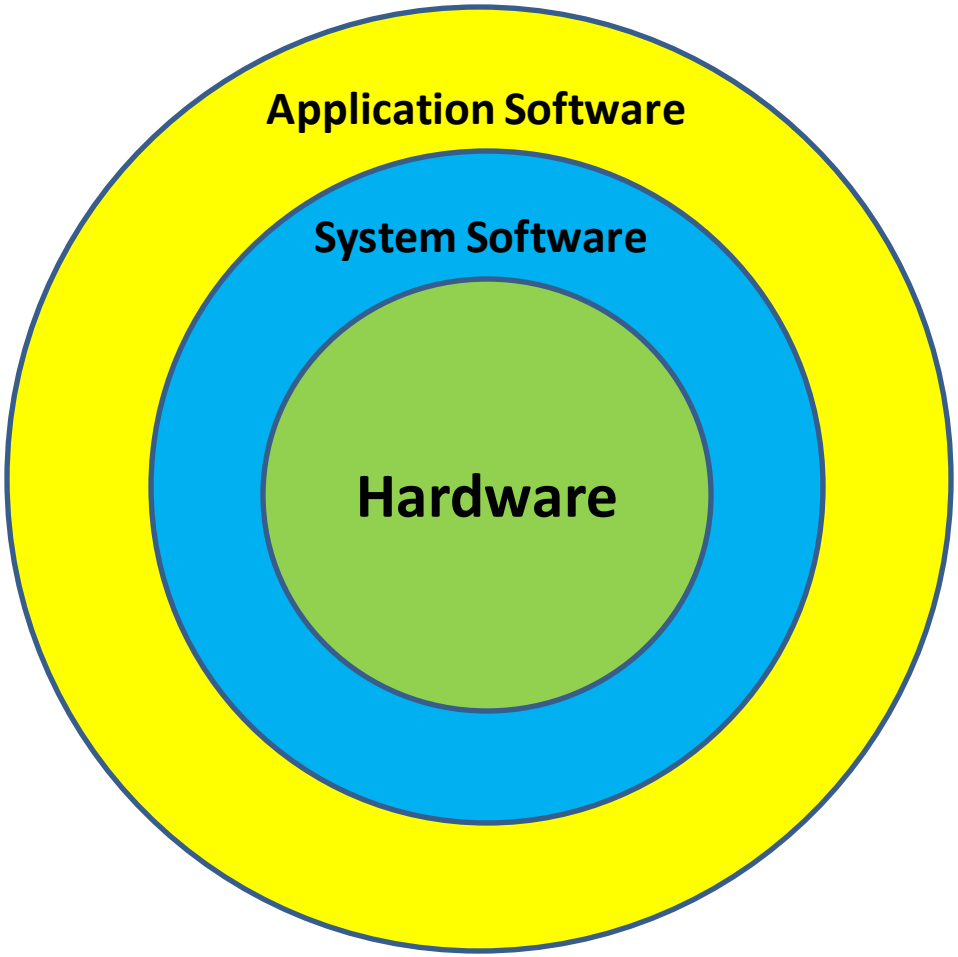


Figure 3

Software Concept

- Hardware devices need user instructions to function.
- A set of instructions that achieve a single outcome are called program or procedure.
- Many programs functioning together to do a task make a software.

Categories of software:

- System software
- Application software
- Utility software

System Software

- Software required to run the hardware parts of the computer and other application software.
- Acts as interface between hardware and user applications
- Interface is needed because hardware devices or machines and humans speak in different languages.
- English is the pre-dominant language of interacting with computers.
- System software converts all human instructions into machine understandable instructions.

Types of system software:

- Operating system
- Language processor
- Device drives

Types of System Software continued…

Operating system

- Operating system is the lifeline of computer.
- OS is the first software that is loaded into the computer memory.
- Computer does not start unless it has an operating system installed in it because OS-

✓ Keeps all hardware parts in a state of readiness to follow user instructions.

✓ Co-ordinate between different devices.

✓ It schedules multiple tasks as per priority.

✓ It allocates resources to each tasks.

✓ It enables the computer to access the network.

✓ Enables users to access and use application software.

Types of System Software continued…

Language processor

It converts all user instructions in machine understandable language.

Three types of languages:

1. Machine level language: a string of 0's and 1's that the machine can understand.
2. Assembly level language: defines mnemonics
3. High level language: uses English like statements and is completely independent of machines.

Types of System Software continued…

Three types of language interpreters:

a. Assembler: Converts assembly level program into machine level program
b. Interpreter: Converts high level program into machine level program line by line.
c. Compiler: Converts high level programs into machine level programs at one go rather than line by line.

Types of System Software

Device driver

- System software that controls and monitors functioning of a specific device on computer.
- Each device that needs to be attached externally to the system has a specific driver associated with it.
- When you attach a new device, you need to install its driver so that the OS knows how it needs to be managed.

Application Software

A software that performs a single task.

Commonly used application software are:

- Word processing
- Spread sheets
- Presentation
- Database management
- Multimedia tools.

Utility Software

- Application software that assists system software in doing their work is called utility software.

Example

- Antivirus software
- Disk Management tools
- File Management tools
- Compression tools
- Backup tools

Assembler

- Assembler is a system software that converts assembly level programs to machine level code.

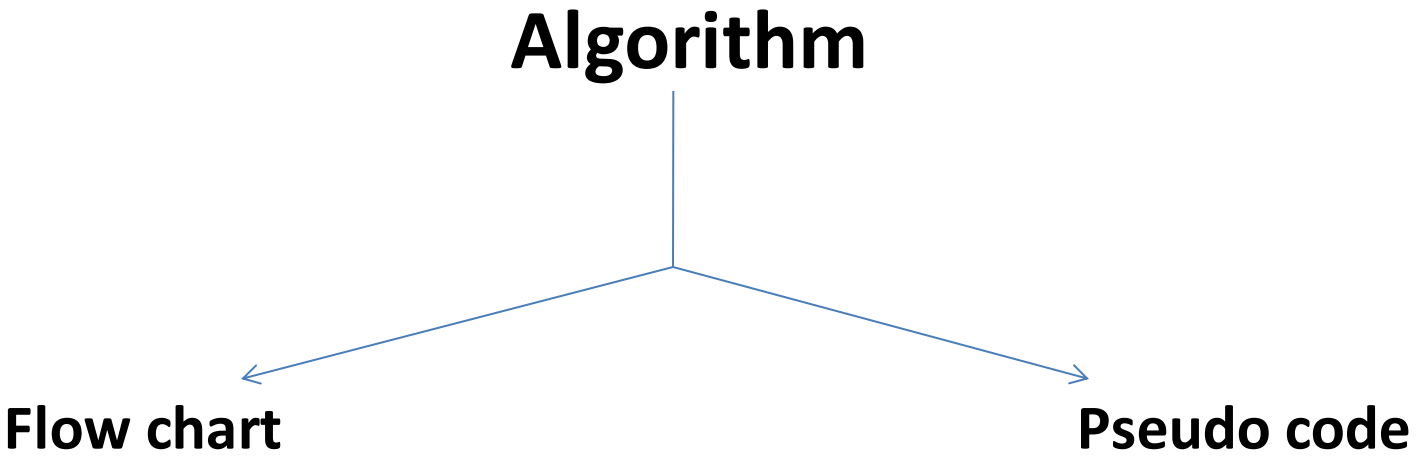
- Advantages provided by assembly level programming:
  - Increases the efficiency of the programmer
  - Productivity increases
  - Programmer has flexibility in writing programs customized to the specific computer.

Compiler and interpreters

- Compilers and interpreters are programs that help convert the high level language (source code) into machine code to be understood by the computers.

Interpreter	Compiler
Translates just one statement of the program at a time into machine code.	Scans the entire program and translates the whole of it into machine code once.
Takes very less time to analyze the code	Takes a lot of time to analyze the source code.
Does not generate an intermediary code. Hence, is highly efficient in terms of memory	Always generates intermediary object code. It needs further linking. Hence more memory needed.
Keeps translating the program continuously till the first error is confronted.	Generates the error message only after it scans the complete program.
Used by programming languages like Ruby and Python for example.	Used by programming languages like C and C++ for example.

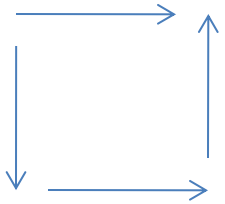
Algorithms



Flowchart

A flowchart is a type of diagram representing a process using different symbols containing information about steps or a sequence of events.

Flowchart symbols

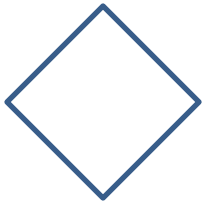


Flow lines

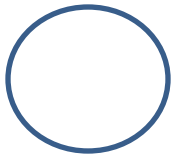
Terminal

Input/output

Processing



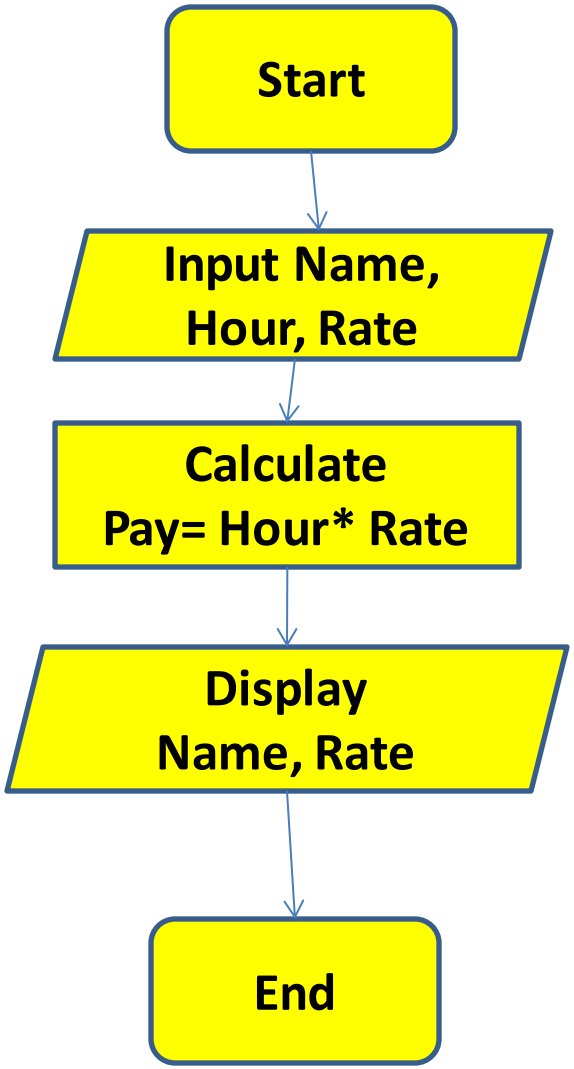
Decision



Connector

Flowchart

Example:



Note: Name, hour and pay are variables in the program

Algorithm

- It is defined as a finite sequence of explicit instructions that, when provide with a set of input values, produces an output and terminates.
- We can use English like phases to describe an algorithm.
- In this case the description is called pseudo code.

Algorithm Example

- Input the three values into the variables Name, Hour, Rate.
- Calculate Pay=Hour*rate
- Display Name and Pay.

Example

Algorithm and flowchart for addition of two numbers.

Algorithm

Step 1: Start

Step 2: Declare variables num1,num2 and sum.

Step 3: Read values for num1, num2.

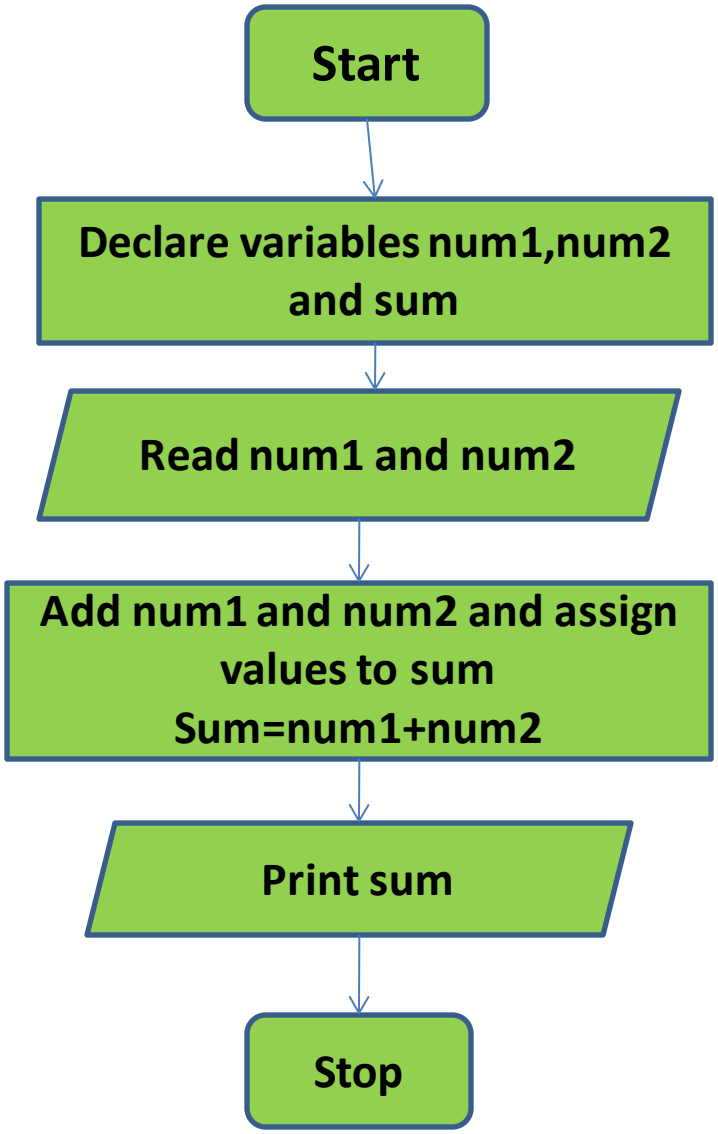
Step 4: Add num1, num2and assign the result to a variable sum

Step 5: Display sum

Step 6: Stop

Example

Flowchart



Source code

- C is a general purpose programming language
- By using the C programming language, we can write the instruction of the computer programs.
- Instructions written in any programming language in human readable form is called as source code.

Source file

- Source file is the file which contains the source code written in C programming language , then that file is also called as the C source file.
- To write the code we can use any text editor and save that as a computer file.
- While writing C programming language it provides a lot of built in features to implement common operations that we have to use in the program.

Source file continued..

- For example: in programming someone has to perform frequently the input operations. And the programmer have to work on distinct value, so for all this operations, the C programming language provides the built in features.
- This features are provided in the form of standard library, which contains all the built in features that will perform the common operations.

Source file continued..

How can we differentiate a file which contains a C programming language from any other file?

For example: if we have a text file we will have a .txt function, .mp3 for music file and .pdf, similarly .C for C source file.

Compilation

- When a program is written using C language, it cant be directly executed in the computer. This is because the computer only understands the code which is written in the binary language.
- That is why the source code needs to be converted to the machine code using compilation process.
- Code generated by the compilation process is called object code.

Compilation continued…

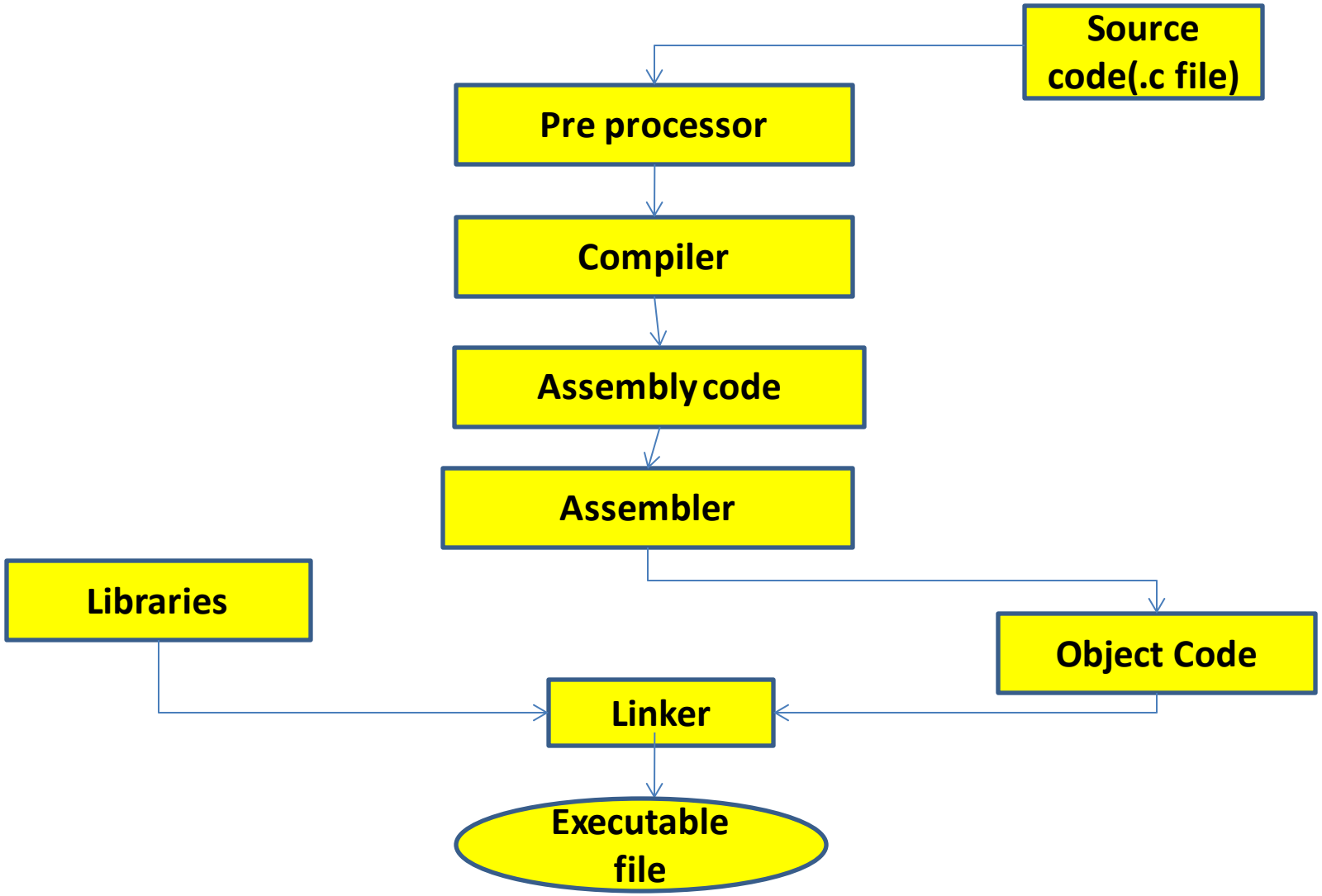
- The file containing the object code is called as the object file.

Represented as:

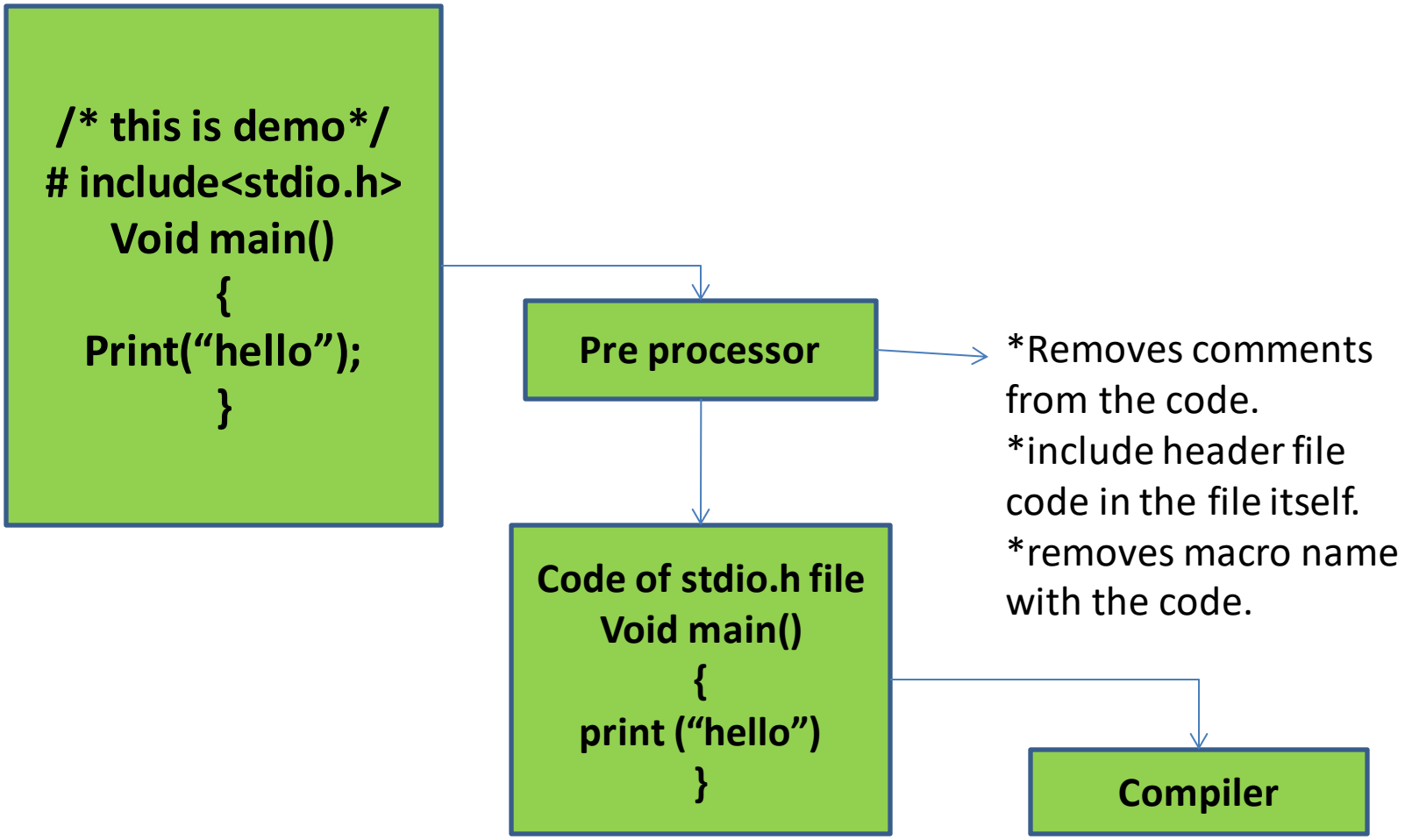
.obj file extension(windows)

.o file extension (linux)

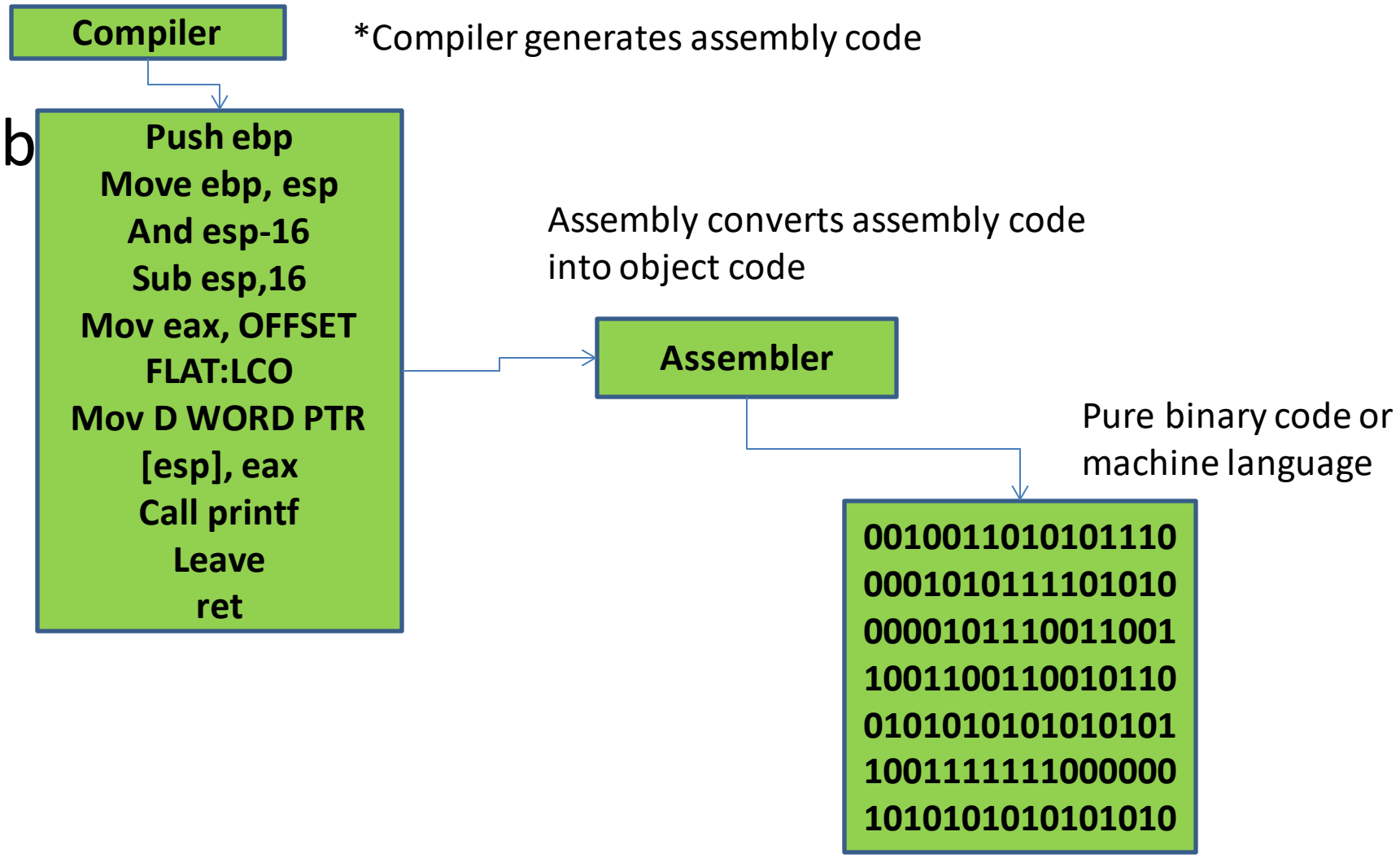
C Compilation process



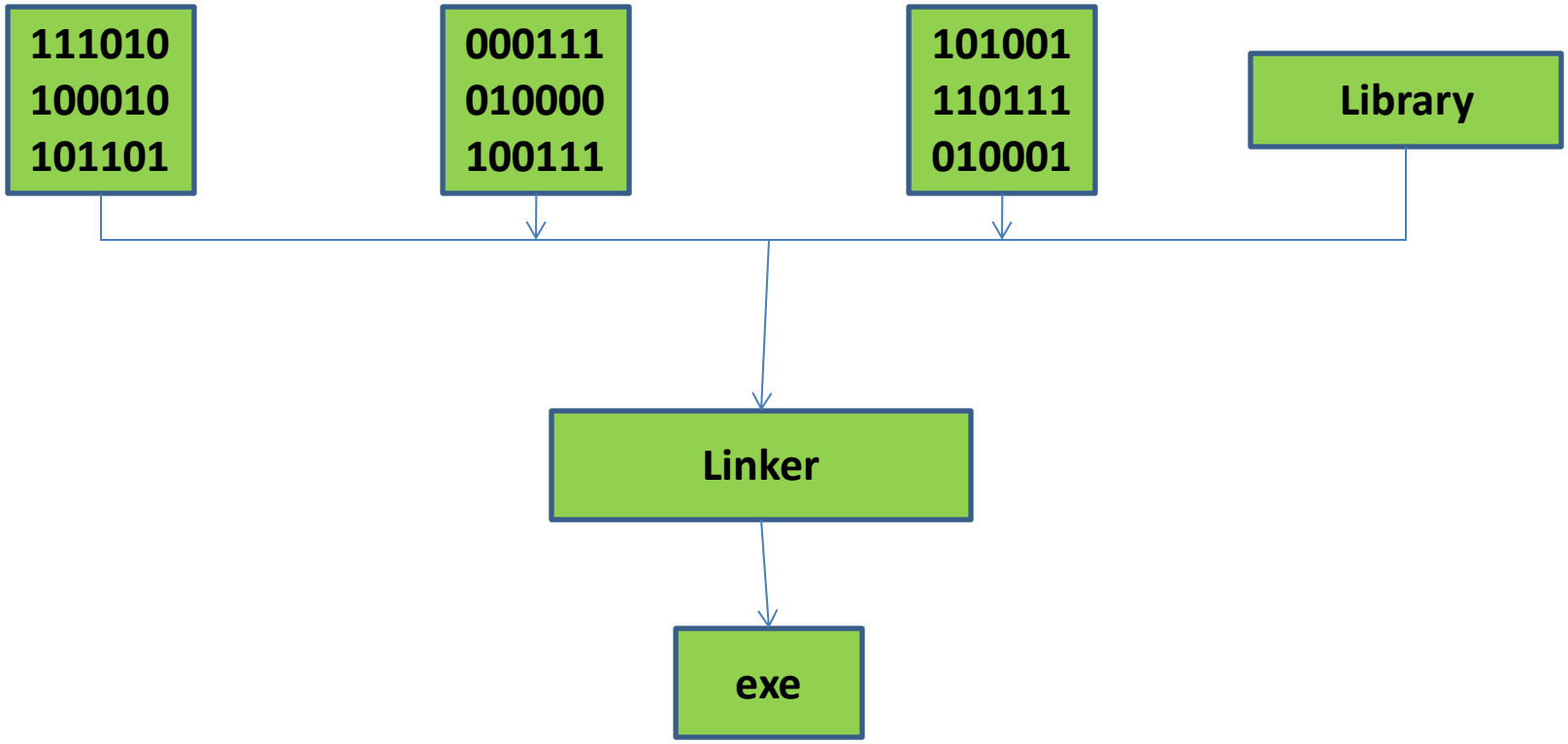
Pre processor



Compiler


Compiler
*Compiler generates assembly code
b
Push ebp
Move ebp, esp
And esp-16
Sub esp,16
Mov eax, OFFSET
FLAT:LCO
Mov D WORD PTR
[esp], eax
Call printf
Leave
ret
Assembly converts assembly code
into object code
Assembler
Pure binary code or
machine language
0010011010101110
0001010111101010
0000101110011001
1001100110010110
0101010101010101
1001111111000000
1010101010101010

Linker


111010
100010
101101
000111
010000
100111
101001
110111
010001
Library
Linker
exe

Linker

- If you are working on a project in which there are multiple modules, at that time we will have a number of C files and when we compile them we will get number of object files.

- We have to merge all the object file into a single one.

- Linker will club all the files, link with the library function, pack everything in a single file and that file is known as .exe file

3 types of Errors

1. Syntax error
2. Runtime error
3. Logical errors

Syntax errors: Errors in our code that the computer cannot interpret. These errors are often

- Spelling errors
- Omission of important characters(such as missing colon)
- Inconsistent use of wrong indentation.

3 types of Errors

Runtime errors: Errors that are not detected until runtime. These are often caused by:

- It cant find some data because it does not exists.
- It cant perform an action on the data it has been given because it is an invalid type of data

Ex: multiply Dog with 300.

3 types of Errors

Logical errors: errors in the code that do not throw an error at all, but simply do not do what you intended the code to do.

**These are the most difficult to spot because they can only be found through full and extensive testing.*

- Temp=int(input("How warm is it today?"))

  if temp<30

  print ("it is hot")

  else

  print("it is cold")

Variables

Definition: variables are simply names used to refer to some location in memory-a location that holds a value with which we are working.

- One whose value can be changed during the execution of the program.
- Instead of entering data directly into a program, a programmer can use variables to store the data in memory.
- The memory location can be identified by the variable name.

Variables continued…

- When the program is executed, the variables are replaced with data.
- The variables must be declared before they can be used in a program.

i. Declaring variables in C:

The syntax for declaring variables in C is:

To declare one variable:

data-type variable;

Example: int a;

float rate;

char ch;

Variables continued…

Declaring of constants: A constant can be declared like a normal variable, but preceded with “const” keyword and a value should be assigned.

for example:const int rate=10, const floatpi=3.14; const ch=‘a’

Variables continued…

To declare more than one variable:

data-type variable1,variable2,…….variablen;

Example: int a, age, sum;

float salary, average;

While declaring a variable, it can be initialized with a value.

For example, int age=18; float rate=10.5; char ch= ‘a’;

When a variable is declared, a memory location gets allocated for the variable and the location holds the value of that variable.

For example: int age=18;

For the variable “age”, two bytes of memory will be alocated and the location holds the value 18

age- name of the variable

18- Value of the variable

Data types

Definition: Data types are declarations for memory locations or variables that determine the characteristics of the data that may be stored and the method of processing that are permitted involving them.

- A data type is to identify various types of data, such as real, integer or character that determines the possible values for that type.
- The data type supported by C are tabulated in table.1

Data types continued…

Types and Description
Basic Types: Three basic types are (a) integer (b) character (c) floating-point
Derived types: They include (a) Pointer (b) Array (c) Structure (d) Union (e) Function
User-defined types: They include (a) typedef (b) enum
The type void: The type specifier void indicates that nothing is available.

Table 1

Structure of C program

The structure consists of the following parts:

- Preprocessor directives
- Variable and function declarations
- Main function
- Other functions.

Preprocessor directives

Structure of C program continued….

Preprocessor directives

- At the time of compilation itself some processes can be done in C.
- The commands which invokes such processes are called preprocessor directives.
- These can optionally be present in the program.
- There are three types of preprocessor directives available with C.
- They are inclusion (#include), macro substitute (#define) and conditional (#if) directives.

Structure of C program continued….

Variables and function declarations

- Outside the main function, variables and functions can optionally be declared.
- Such variables and functions declared before the function main() are global and hence are available to all functions.

Structure of C program continued….

Main function (main())

- This is the main function of any C program.
- C compilers starts execution from this main ().

Structure of C program continued….

Other functions

- All other functions called from main() or any other functions can be present after the statements of main().
- Though this sequence is followed always, C compilers allows to have other function statements to appear before the main program. But still the C compiler starts compiling from main()

Structure of C program continued….

Example 1

```
/* structure of C language */
/* Preprocessor Directives */
#include <Math.h>
#include <Stdio.h>
#define MAX 20
/*variable and function declarations */
int val,tot;
float sum_dig(sum);
/* main function and its statements */
main()
{
   int a,b;
   printf("Enter two values → ");
   scanf("%d %d",&a,&b);
   printf("The value of MAX is fixed as %d\n", MAX);
   if (a > b)
      printf("big is %d\n",a);
   else
      printf("big is %d\n",a);
   printf("sum of the digits of %d is %f\n",
      a,sum_dig(a));
   printf("sum of the digits of %d is %f\n",
      b,sum_dig(b));
}
```

Structure of C program continued….

Example 2 (a)

```
main()
{
   int a,b;
   printf("Structure of C language !!");
   a = 5;
   b = a * 12;
}
```

The above program can be presented as given below which is also correct.

Example 2 (b)

```
main() {printf("Structure of C language !!");a
          = 5; b = a * 12; }
```

Example 2 (c)

```
main() {printf("Structure of C language !!");
a = 5; b = a * 12; }
```

But, 2 (b) and 2 (c) are lacking the readability. Hence it is always advisible to present the program with readability as in 2(a). This makes the debugging and understanding of the program easier.