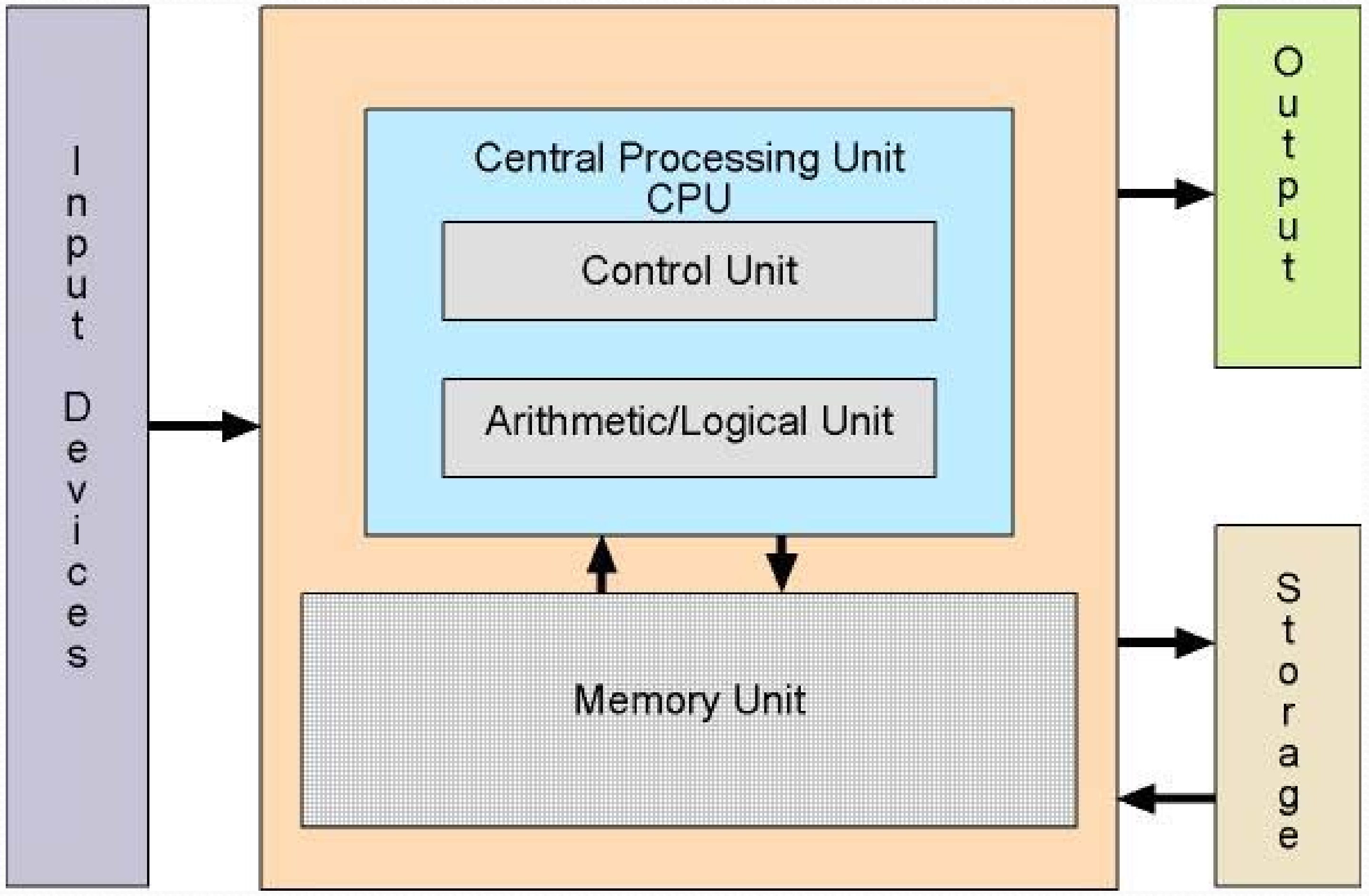
Introduction to C Programming

Problem Solving Through ‘C’

- Learning ‘C’ Programming
  - Syntax – set of rules that defines the grammar of the programming language. (Checking symbolic representation).
    - A **x** B (language L1: multiplication) **[Error in L2]**
    - A ***** B (language L2 : multiplication) **[Error in L1]**
  - Semantics – set of rules used to derive the meaning of the statements.
    - In language L3
      - A **x** B (multiplication)
      - A ***** B (power)
  - Solving

$\xrightarrow{}$ Given to find multiplication of two numbers. If you write **A*B** then semantics is not correct.

Block Diagram of a Computer

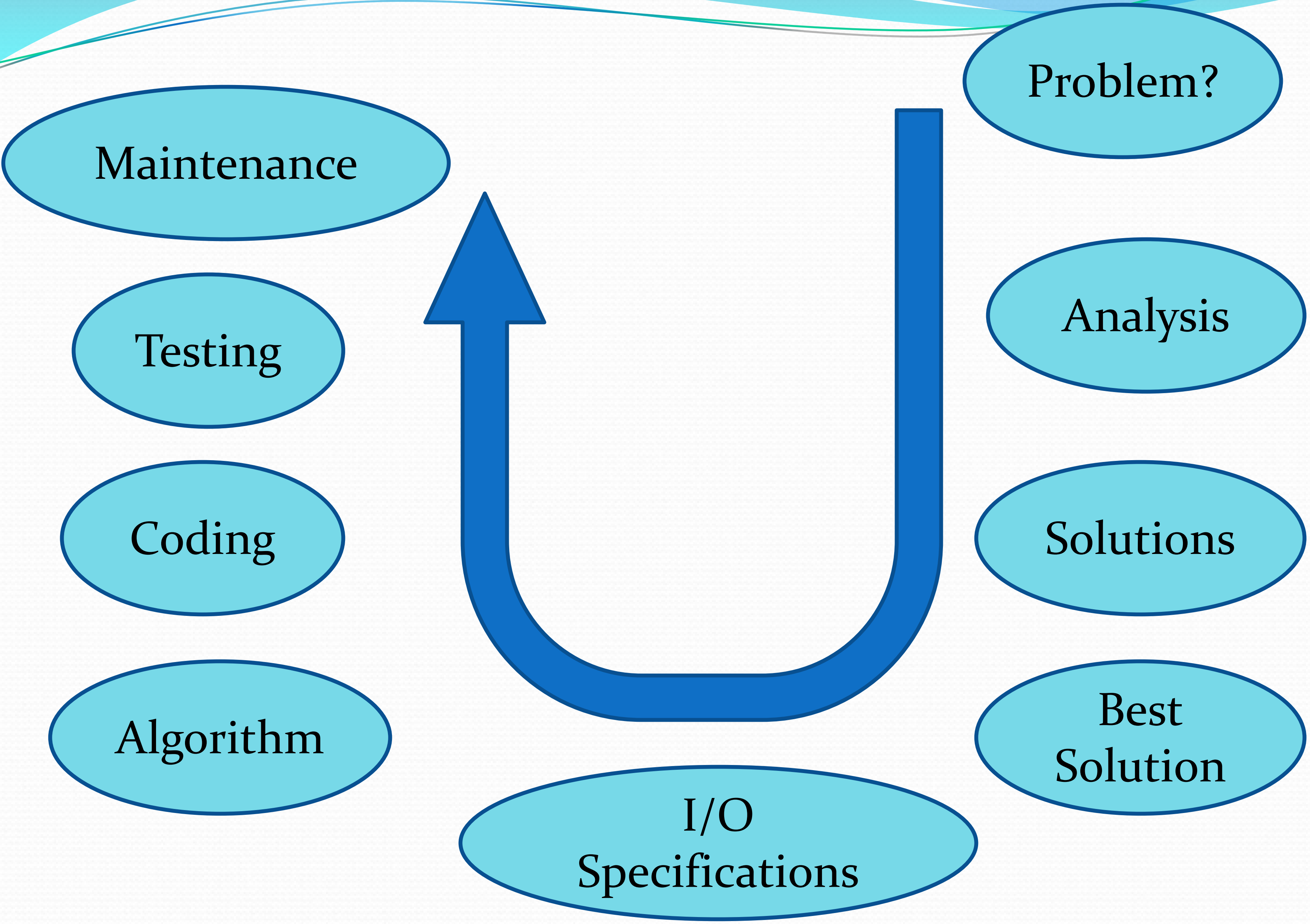


Types of Real World Problems

- In real world applications we come across day to day problems
  - Going to market for shoping
  - Solving quadratic equations
  - Matrix Operation: Add, Multiply...
  - ATM
  - Grade calculation
  - Railway/Flight Reservation
  - Library Automation
  - Many More.....

Approach to Problem Solving

- Define and Identify the Problem
- Analyze the Problem
- Identifying Possible Solutions
- Selecting the Best Solutions
- I/O Specifications
- Algorithm
- Coding
- Testing
- Maintenance


Problem?
Analysis
Solutions
Best Solution
I/O Specifications
Algorithm
Coding
Testing
Maintenance

An Example- Quadratic Equation

- Problem Definition: A quadratic equation of the form

$$ax^2 + bx + c = 0$$

- Analysis: The equations helps to find
  - Two values x_1 and x_2
  - Given constants a, b, c
  - Value of $a \neq 0$

- Solutions:
  - Factorization
  - Quadratic Formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

- Best Solution
  - Choose a best possible solution

- I/O:
  - Input: a, b, c
  - Output: x_1 and x_2

- Algorithm/Flowchart:
  - An algorithm is a sequence of well defined steps often used for solving problem
  - A flowchart is a schematic representation of an algorithm

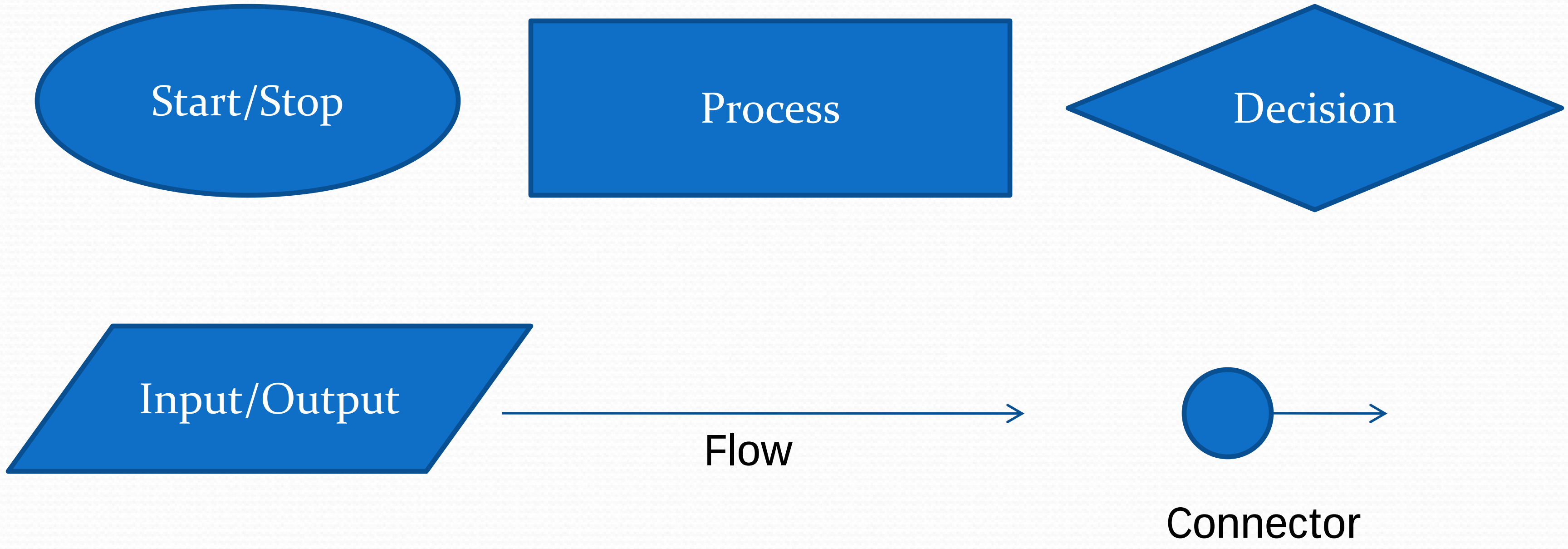
- Coding
  - Set of instructions for solving a problem
- Testing
  - Finding the errors in the code for possible values
- Maintenance
  - Modification due to requirements change
  - Scalability

Introduction to Algorithms

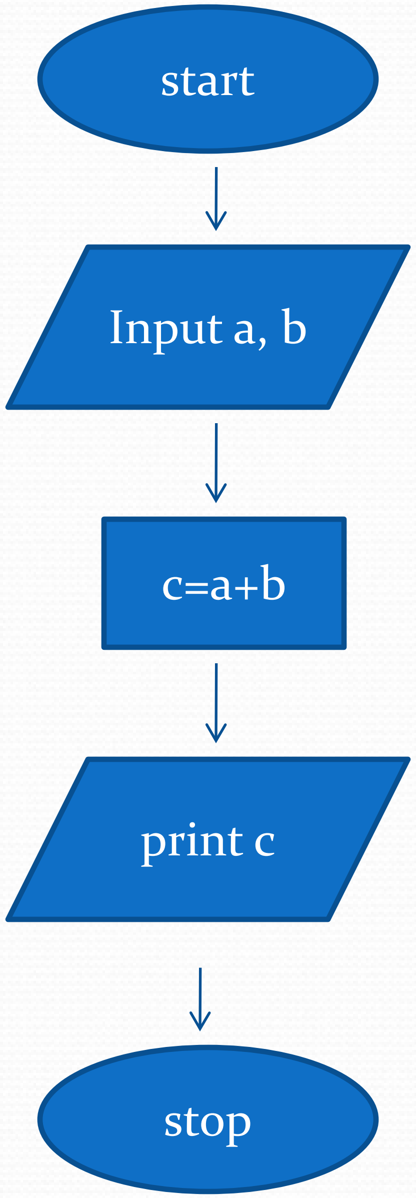
- Step by step method to solve a problem
- Must include ALL required information

Flowcharts

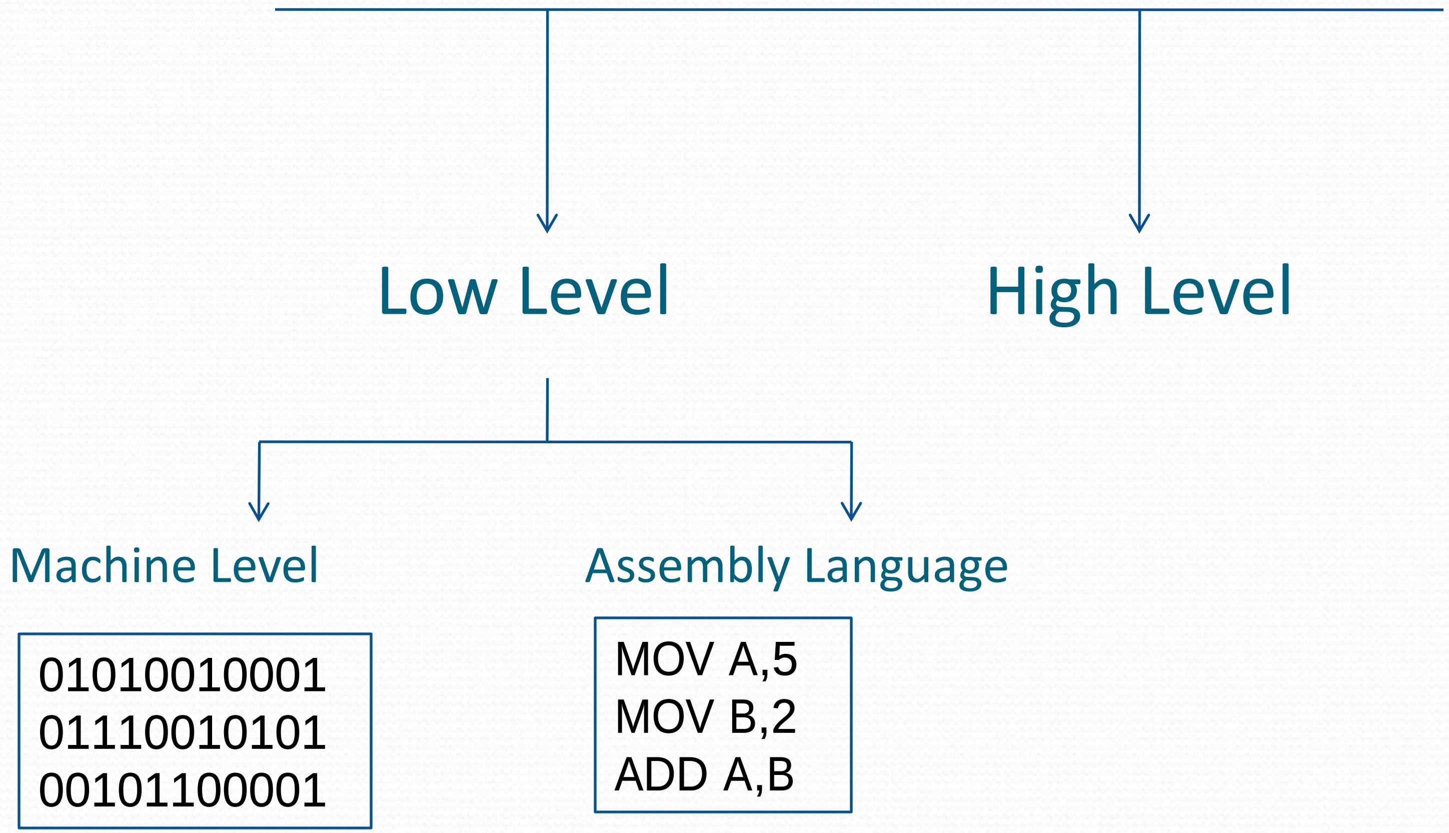
- Graphical Representation of problems
- Symbols



Add Two Numbers



Programming Languages



```
01010010001
01110010101
00101100001
```

```
MOV A,5
MOV B,2
ADD A,B
```

More About High Level Languages

- The languages are designed keeping in mind features of portability
- The languages are machine independent
- Easy to write and understand
- The programmer pays whole attention logic of the program

Translators

- For translating high level and Assembly Language to machine language
  - Assembler: Assembly Level $\longrightarrow$ Machine Level
  - Interpreter
    - High Level $\longrightarrow$ Machine Level
    - Interpreter searches the error statement by statement
  - Compiler
    - High Level $\longrightarrow$ Machine Level
    - Compiler searches all errors in the code and lists them

Some High Level Languages

- BASIC
- FORTRAN
- PASCAL
- COBOL
- C
- C++
- Java

C

- Developed at AT&T Laboratory of USA
- Year: 1970
- Created By: Dennis Ritchie

$$\begin{tikzcd}
\text{BCPL} \arrow[d] \\
\text{B} \arrow[d] \\
\text{C}
\end{tikzcd}$$

(Basic Combined Programming Language)

Why C?

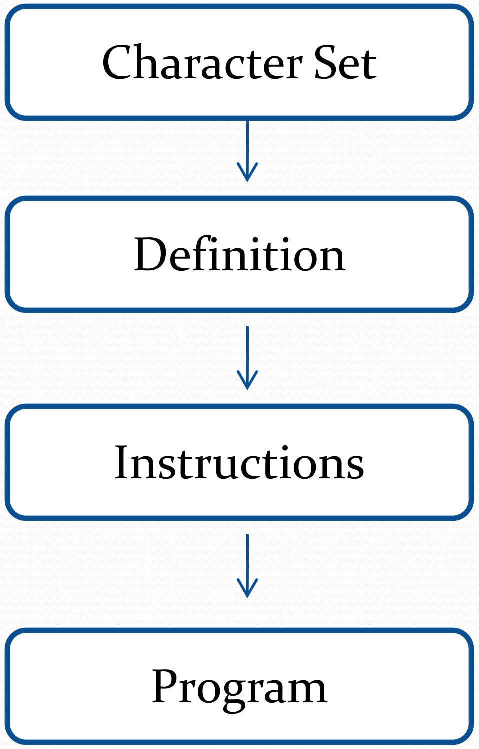
- A basic foundation for learning programming language elements

- Major parts of popular OS like windows, Linux, Unix are written in C

- Embedded System Programs are written in C
  - Microwave Oven
  - Washing machines
  - Digital Cameras

- C provides several language elements that makes interaction with hardware
- Several Gaming programs are also developed in C

Getting Started



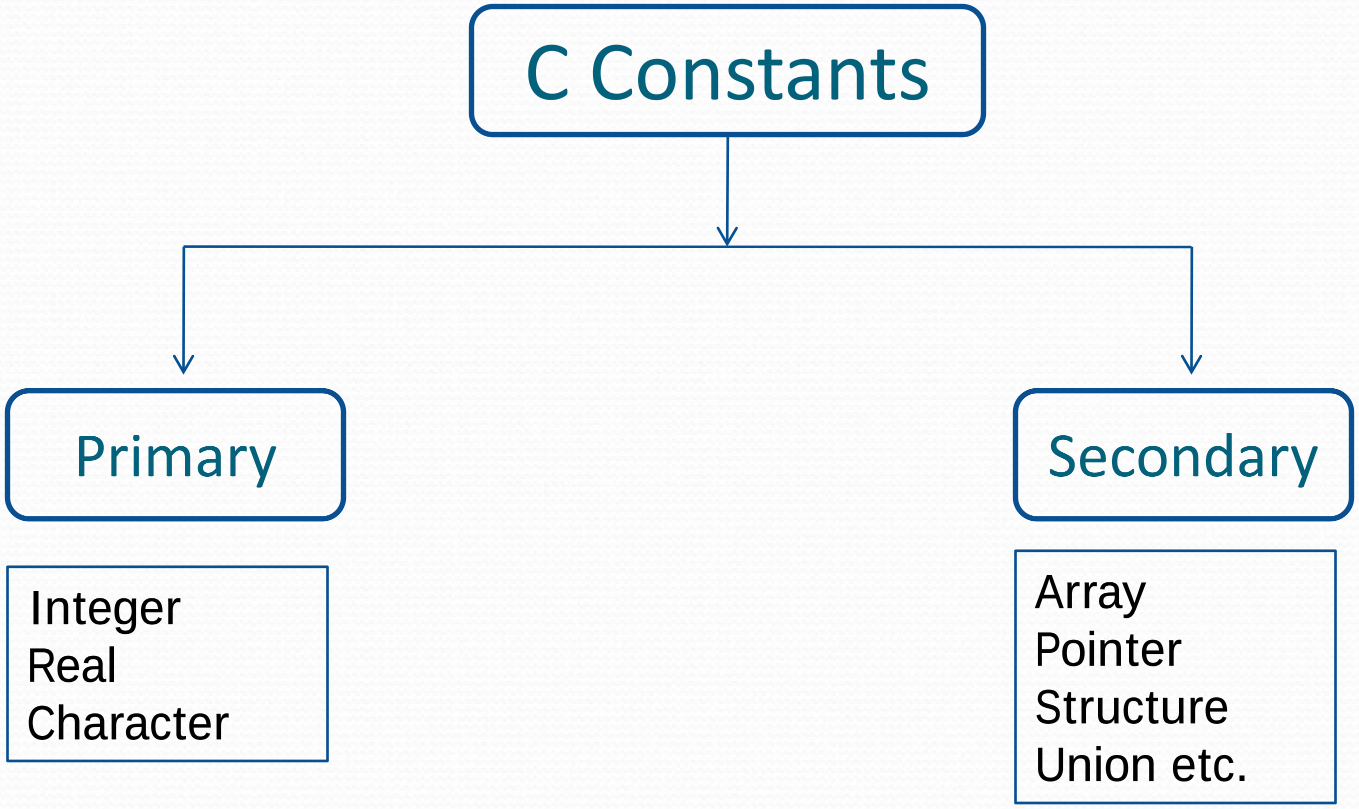
A, B, C……….Z
a, b…..z
0,1,2………9
@,*,&……….

Constants
Variables
Keywords

Syntax

```
#include<stdio.h>
main()
{
printf("Hello World")
}
```

Types of C Constants



First C Program

```
// single line comment
/* multiple line
  comments within code*/
#include<stdio.h>
main()
{
printf(“Hello World\n”);
}
```

To include information about standard input/output

Function main

Body of Function