

# C++ LAUNCHPAD



## Lecture-2 FUNDAMENTALS - |

- Programming Fundamentals in C++

Rajesh Sachdev

## BT - 1: Hour Glasses

You have two hourglasses: a 7 minute one and a 11 minute one. Using just two hourglass, accurately time 15 minutes.

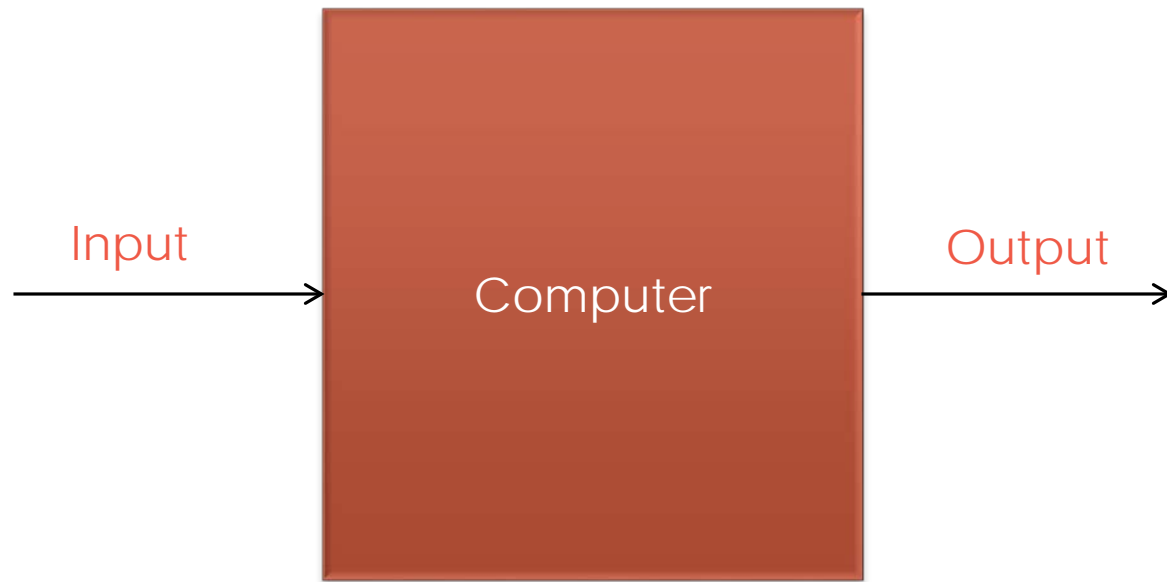
## BT – 2: Apples and Oranges

There are three closed and opaque cardboard boxes. One is labeled "APPLES", another is labeled "ORANGES", and the last is labeled "APPLES AND ORANGES". You know that the labels are currently misarranged, such that no box is correctly labeled. You would like to correctly rearrange these labels. To accomplish this, you may draw only one fruit from one of the boxes. Which box do you choose, and how do you then proceed to rearrange the labels?

## BT – 3: Average Salary

Three coworkers would like to know their average salary. However, they are self-conscious and don't want to tell each other their own salaries, for fear of either being ridiculed or getting their houses robbed. How can they find their average salary, without disclosing their own salaries?

# What does a computer do?

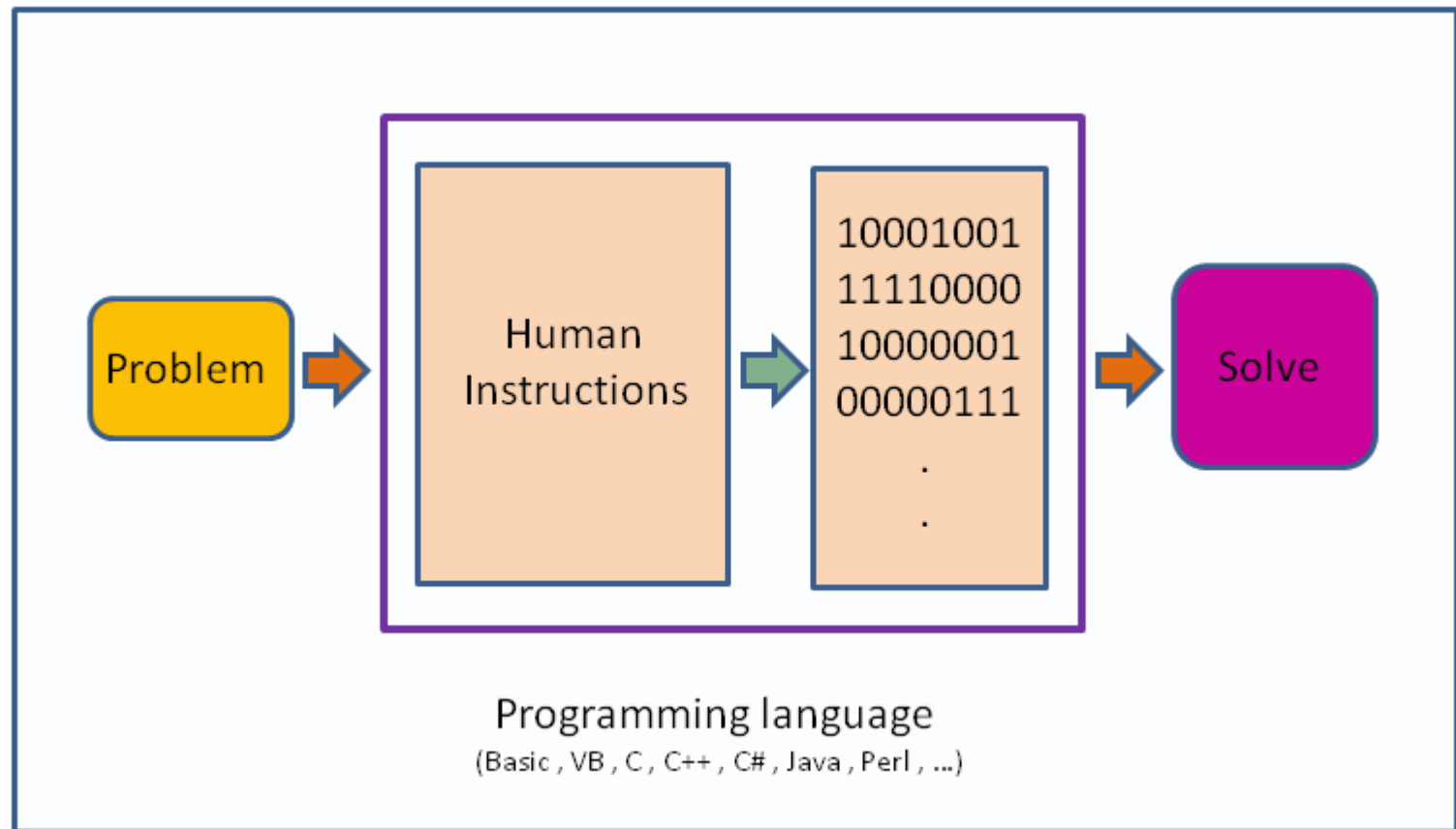


# Tool for solving problems with data

- To communicate the solution we create Programs.
- So a program is a sequence of instructions that tells a computer how to perform a task.
- When computer follows the instructions we say it executes the program.

# It's a machine!

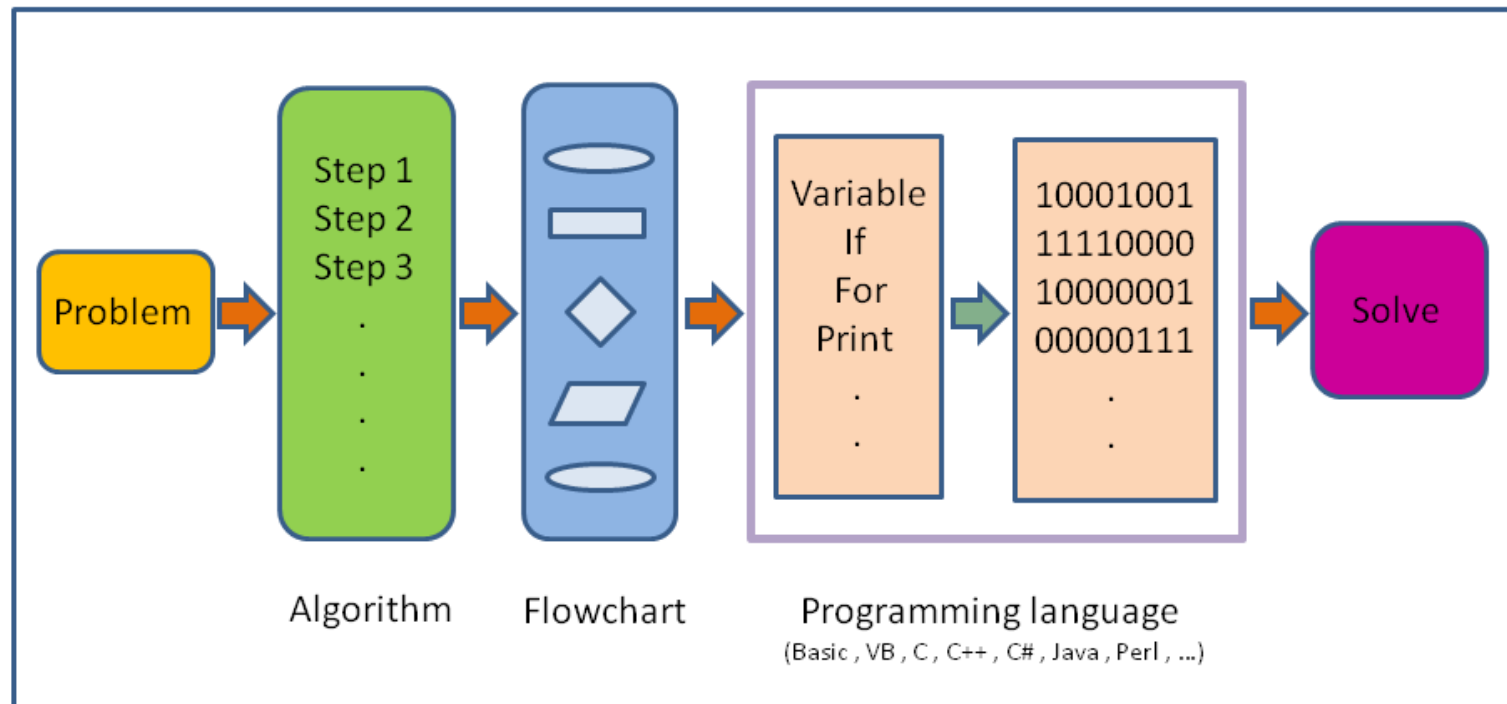
- Computers are a machine, and at the most basic level, they are collection of switches – where 1 represents “on” and 0 represents “off”.
- Everything that a computer does is implemented in this most basic of all numbering systems – binary
- So if you want to communicate directly we need to talk to in binary





Its nearly impossible to write  
in Binary!

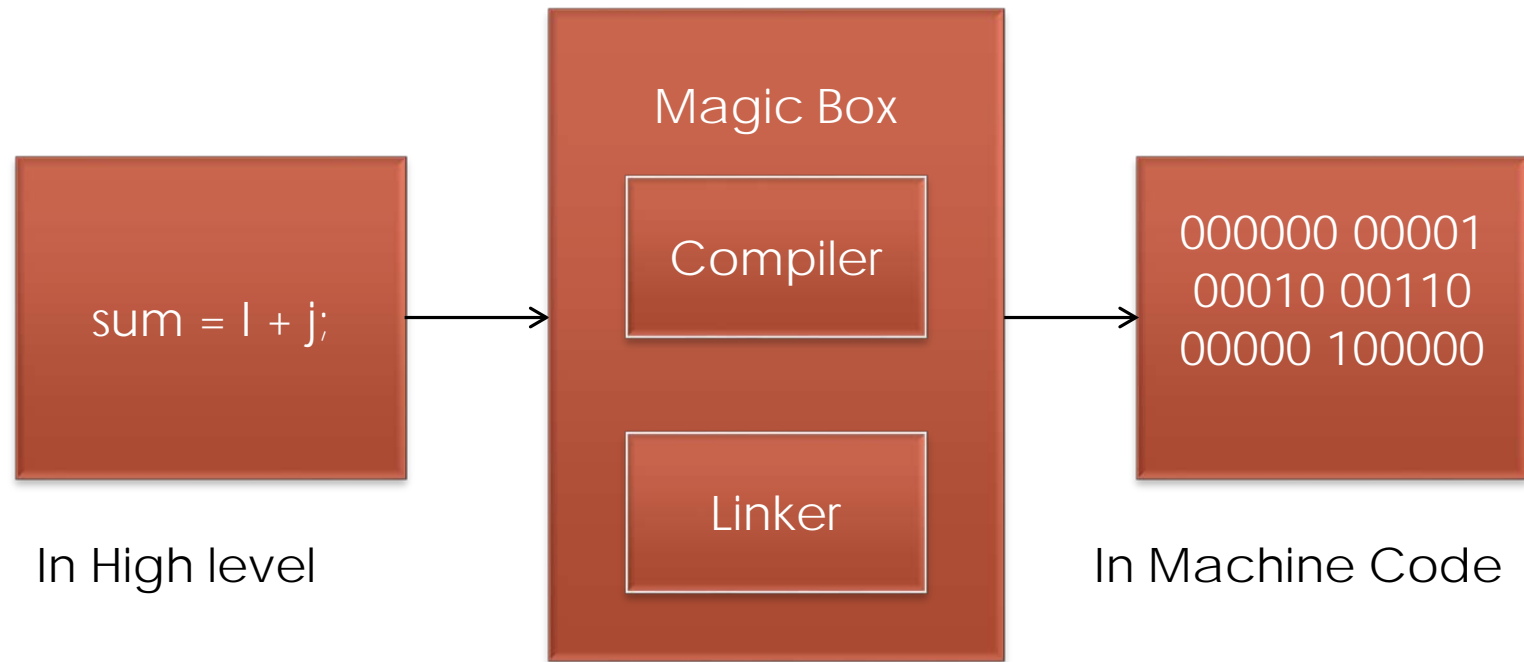
So we use programming  
language with Flowcharts  
& Algorithms for solving a  
Problem



# What is programming language?

- ◉ A programming language is a set of rules that provides a way of telling a computer what operations to perform
- ◉ It provides a linguistic framework for describing computations.
- ◉ A programming language also has words, symbols and rules of grammar.
- ◉ The grammatical rules are called syntax

# How do we work with High Level?



Before we write a program  
for a solution we need an  
Algorithm.

# So what is an algorithm?

- An algorithm is a self-contained step by step set of operations to be performed in order to solve a problem.
- Its an effective method that can be expressed within a finite amount of space and time and in well-defined formal language for solving a problem.
- Another way to describe an algorithm is a sequence of unambiguous instructions.

# Expressing Algorithms?

- Algorithms can be expressed in many kind of notations, including natural languages, pseudocode, flowcharts, etc.
- Natural Language expressions of algorithms tend to be verbose and ambiguous, and are rarely used for complex or technical algorithms.
- Programming languages are primarily intended for expressing algorithms in a form that can be executed by a computer.



# Two basic aspects of programming

- Data
- Instructions

To understand data we  
need to understand  
Variables!

# What are Variables?

- Variables in a computer program are analogous to Buckets or Envelopes where information can be maintained and referenced.
- On the outside of the bucket is a name.
- When referring to the bucket, we use the name of the bucket, not the data stored in the bucket.

# Variable Actions!

- Create one (with a nice name). A variable should be named to represent all possible values that it might contain. Some examples are: `midterm_score`, `midterm_scores`, `data_points`, `course_name`, etc.
- Put some information into it (destroying whatever was there before). We "put" information into a variable using the assignment operator, e.g., `midterm_score = 93;`
- Get a copy of the information out of it (leaving a copy inside). We "get" the information out by simply writing the name of the variable, the computer does the rest for us, e.g., `average = (grade_1 + grade_2) / 2.`

# Now lets talk about Instructions!

September 30, 2017



# What are instructions?

- Its an order given to computer.
- At lowest level each of these is a sequence of 0s and 1s.
- In assembly language, each statement is one instruction but in high level each statement can be further broken into multiple steps.

# Six basic computer instructions

- Reading/Receiving some information
- Outputting/Printing some information
- Performing arithmetic operation
- Assigning a value to a variable or memory location
- Conditional Execution
- Repeat a group of actions

# How is data stored?

- ◉ Whole Numbers / Positive Integers – Binary Equivalent. E.g: 23 → 10111
- ◉ Characters – Binary Equivalent of their ASCII Values. E.g: 'A' → 65 → 1000001
- ◉ Negative Numbers – 2's complement of their counter part
- ◉ Floating Values – Binary Equivalent of 2 integers [ Significant and Exponent]. E.g: 1.23 can be respresented as  $123 * 10^{-2}$ . So we now have two integers 123 and -2.



# Time to write our first program!

# Program to print "Hello World"

- ◉ #include <iostream>
- ◉ using namespace std;
- ◉ int main() {
- ◉     cout << "Hello world!";
- ◉     return 0;
- ◉ }

# Identifiers

- A C++ identifier is a name used to identify a variable, function, class, module, or any other user-defined item
- An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by zero or more letters, underscores, and digits (0 to 9)

# Keywords

- Some reserve words which cannot be used as identifiers
- These are basically part of the grammar representing the language
- E.g: if, while, return, namespace, etc.

# Data types

- As we know we need variables to store information.
- We might want to store information of various types in a variable like character, whole numbers, integers, floating point, boolean etc.
- Based on the data type of a variable, the operating system allocates memory and interprets the combination of 0s and 1s in that memory

# Primitive Data Types

- Boolean - bool
- Character - char
- Integer - int
- Floating Point - float
- Double Floating Point - double

# Data type modifiers

- Several of the basic types can be modified using one or more of these type modifiers
- signed
- unsigned
- short
- long

# Data types – size and range!

Type	Typical Bit Width	Typical Range
char	1byte	-127 to 127 or 0 to 255
unsigned char	1byte	0 to 255
signed char	1byte	-127 to 127
int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295
signed int	4bytes	-2147483648 to 2147483647
short int	2bytes	-32768 to 32767
unsigned short int	Range	0 to 65,535
signed short int	Range	-32768 to 32767
long int	4bytes	-2,147,483,647 to 2,147,483,647
signed long int	4bytes	same as long int
unsigned long int	4bytes	0 to 4,294,967,295
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)



# Variables

- C++ is strongly typed language, so every variable must be defined before using it.
- `type variablelist; // type is the type (eg. int), varName is the name of the variable`
- e.g:
  - `int sum;`
  - `char ch;`
  - `float a, b;`
- Variables when just declared have garbage value until they are assigned a value for the first time
- We can assign a specific value from the moment variable is declared, called as initialization of variable. `[float b = 0.0;]`

# Statements & Expressions

- A statement in C++ is the smallest independent unit in the language
- Statements in C++ are terminated by a semicolon
- An expression is a mathematical entity that evaluates to a value
- Expressions are often used inside of statements.

Lets look at few statements  
and expressions!

# Basic Operators in a Expression

- Unary [ +, - ]
  - Arithmetic [ +, -, /, \*, % ]
  - Brackets [ ( ) ]
  - Assignment [ = ]
  - Relational [ ==, !=, >, <, >=, <= ]
  - Logical Operators [ &&, ||, ! ]
- 
- PS - 1: Relational Operators and Logical Operators always Evaluate to 0 or 1
  - PS – 2: For logical evaluation any non-zero value is true.

# If Block

- Single If
  - if (a > 10) {
  - cout << "Hello!";
  - }
- If Else
  - If (a>10) {
  - cout << "Hello!";
  - } else {
  - cout << "World.";
  - }
- If .. Else If .. Else
  - If (a>10 && a <20) {
  - cout << "Hello!";
  - } else if (a >20 && a <30) {
  - cout << "Hello World!";
  - } else {
  - cout << "Welcome to Coding Blocks";
  - }

# While block

- `while( condition is true ) {`
- `//do some stuff`
- `}`

# Lets convert some pseudocodes!

- Read P,R,T and calculate SI
- Find largest of 3 numbers
- Check if a number is prime or not
- Write a program to print the following pattern
  - 1
  - 2 3
  - 4 5 6
  - 7 8 9 10

# Recap

- Program Always starts with `main()`
- `{ }` are used to enclose a block (function, if, while etc.).
- C++ Compiler Ignores whitespace (space, carriage returns, linefeeds, tabs, vertical tabs, etc.)
- Output using `cout`
- Input using `cin`
- Header Files
- Comments (`//` & `/*...*/`)



## Time to try?

- Given a list of N integers, find mean, maximum and minimum value. You would be given first N, and then N integers of the list.
- Print all prime numbers between 2 to N
- Read N and print the below pattern
  - 1
  - 232
  - 34543
  - 4567654
  - 567898765

# Time for Brain Teasers!

## BT – 5: Circular Jail Cell

- There is a circular jail with 100 cells numbered 1-100. Each cell has an inmate and the door is locked. One night the jailor gets drunk and starts running around the jail in circles. In his first round he opens each door. In his second round he visits every 2nd door (2,4,6---) and shuts the door. In the 3rd round he visits every 3rd door (3,6,9---) and if the door is shut he opens it, if it is open he shuts it. This continues for 100 rounds (i.e. 4,8,12 --- ; 5,10,15 ---; ---; 49,98 etc.) and exhausted the jailor falls down.
- How many prisoners found their doors open
- after 100 rounds?

## BT – 6: Greedy Pirates

- A pirate ship captures a treasure of 1000 golden coins. The treasure has to be split among the 5 pirates: 1, 2, 3, 4, and 5 in order of rank. The pirates have the following important characteristics: infinitely smart, bloodthirsty, greedy. Starting with pirate 5 they can make a proposal how to split up the treasure. This proposal can either be accepted or the pirate is thrown overboard. A proposal is accepted if and only if a majority of the pirates agrees on it. What proposal should pirate 5 make?

## BT – 7: Infinite Quarter Sequence

- You are wearing a blindfold and thick gloves. An infinite number of quarters are laid out before you on a table of infinite area. Someone tells you that 20 of these quarters are tails and the rest are heads. He says that if you can split the quarters into 2 piles where the number of tails quarters is the same in both piles, then you win all of the quarters. You are allowed to move the quarters and to flip them over, but you can never tell what state a quarter is currently in (the blindfold prevents you from seeing, and the gloves prevent you from feeling which side is heads or tails). How do you partition the quarters so that you can win them all?

## BT – 8: Daughters' Ages

- Local Berkeley professors Dr. X and Dr. Y bump into each other after a long time.
- X     hey! how have you been?
- Y     great! i got married and i have three daughters now
- X     really? how old are they?
- Y     well, the product of their ages is 72, and the sum of their ages is the same as the number on that building over there
- Y     right, ok ... oh wait ... hmm, i still don't know
- X     oh sorry, the oldest one just started to play the piano
- Y     wonderful! my oldest is the same age!
- How old are the daughters?

# What is next class about?

# Programming Fundamentals - 1

- Operators/loops
- Continue, break;
- Arrays



# C++ LAUNCHPAD



Thank You !

Rajesh Sachdev

# Warm up!

- Write a pseudo code to print the following pattern
- 1
- 01
- 101
- 0101
- 10101

## BT – 4: Criminal Cupbearers

- An evil king has 1000 bottles of wine. A neighboring queen plots to kill the bad king, and sends a servant to poison the wine. The king's guards catch the servant after he has only poisoned one bottle. The guards don't know which bottle was poisoned, but they do know that the poison is so potent that even if it was diluted 1,000,000 times, it would still be fatal. Furthermore, the effects of the poison take one month to surface. The king decides he will get some of his prisoners in his vast dungeons to drink the wine. Rather than using 1000 prisoners each assigned to a particular bottle, this king knows that he needs to murder no more than 10 prisoners to figure out what bottle is poisoned, and will still be able to drink the rest of the wine in 5 weeks time. How does he pull this off?

# Binary Number System!