IEEE SCT SB

IEEE

Code in 10 days

Day 1

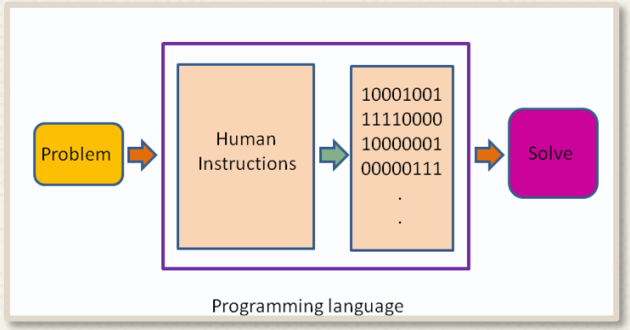
dotSlash
Workshop series

Topics for Today

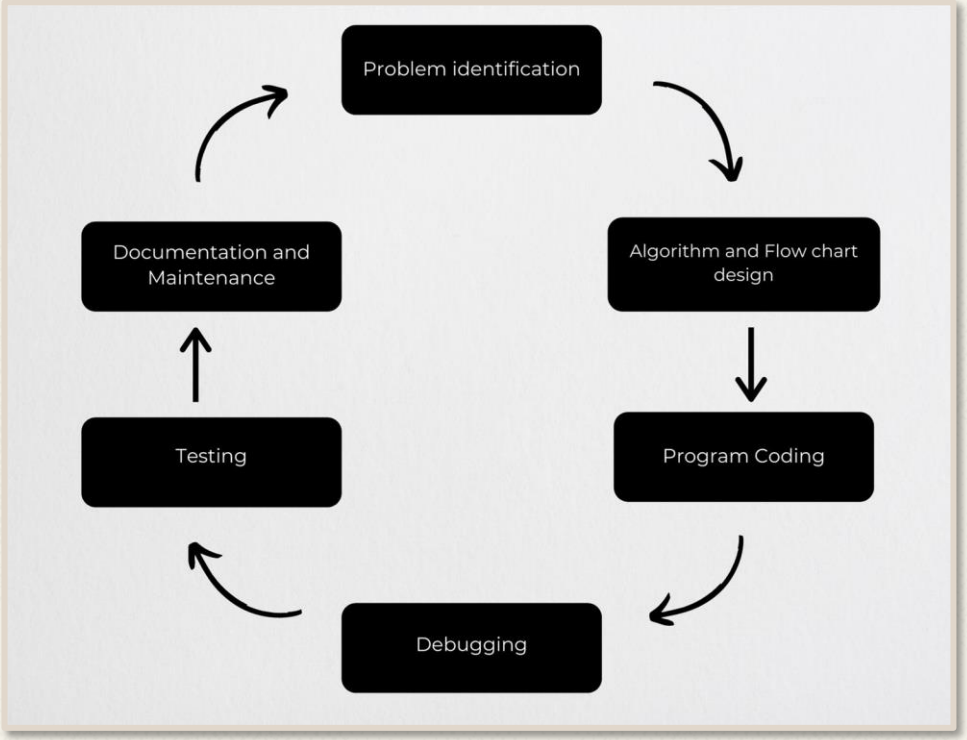
- Introduction to Programming
- About Algorithms and Flowcharts
- Getting started with C++

Programming

- A program serves the purpose of commanding a computer.
- They are step by step instructions to a computer to perform a specific task.
- Programming languages are used to communicate with the computer.



Phases of Programming



Algorithms and Flowcharts

- An algorithm is the step by step procedure to solve a problem.
- Whereas, a flowchart is the pictorial representation of an algorithm.
- Flowcharts have specific symbols for different instructions.

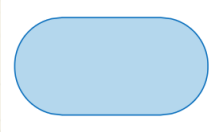
Advantages of Algorithms

- Effective communication.
- Efficient and easy coding.
- Easy debugging.
- Independent of programming languages.

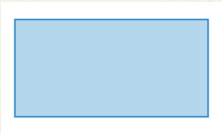
Advantages of Flowcharts

- Effective and easier communication.
- Effective analysis and synthesis.
- Efficient coding.
- Easy debugging and efficient testing.
- Efficient program maintenance.
- Proper documentation.

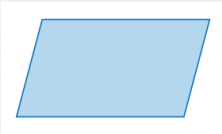
Basic Flowchart Symbols



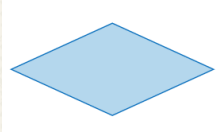
Start/End
- Represents the start/end point of a flowchart



Process
- Represents a process

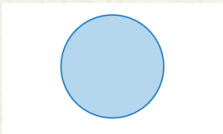


Input/Output
- A parallelogram represents input or output.



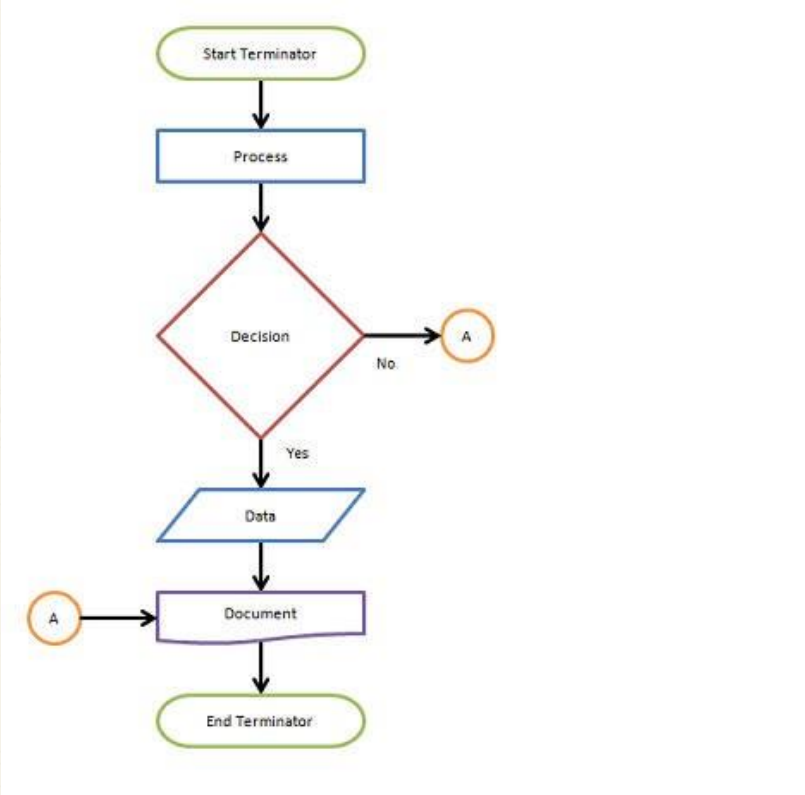
Decision
- Indicates a decision.

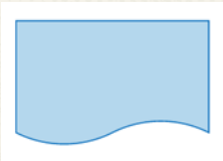
Arrows
- A connector that represents the relationships between the different structures.



Connector
- Indicates that the flow continues where a matching symbol is found.

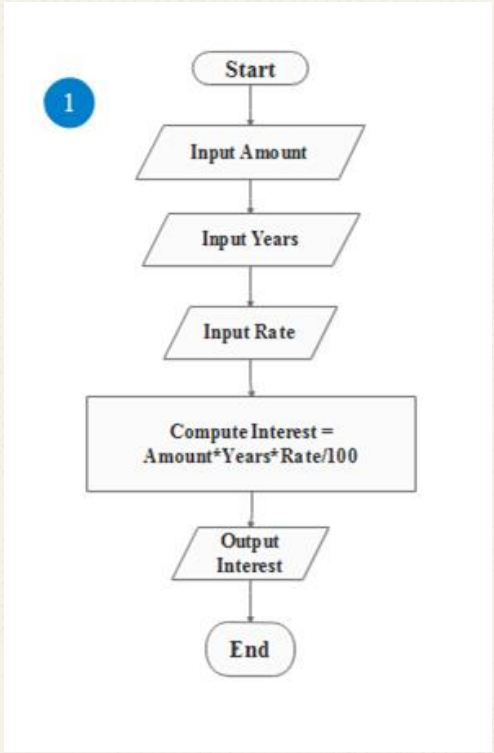
Flowcharts





Document
- Represents a printed document

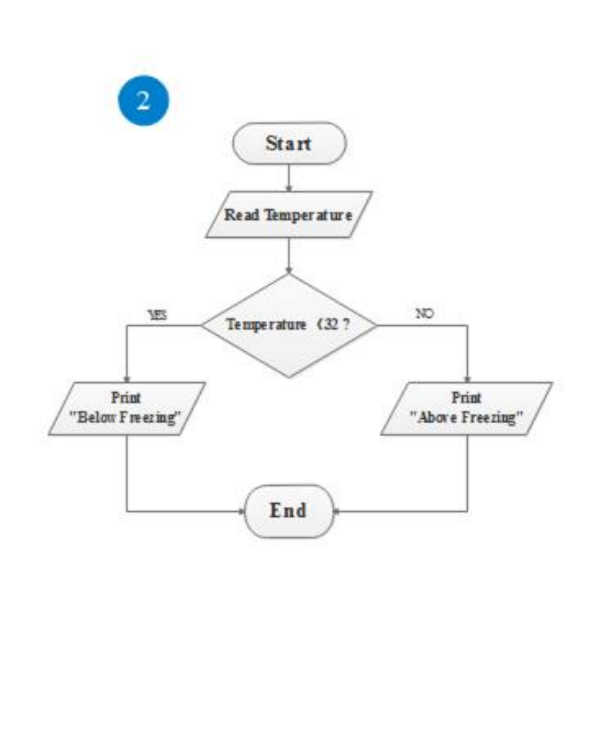
Example



Algorithm

Step 1: Start
Step 2: Input the amount.
Step 3: Input the number of years.
Step 4: Input the interest rate.
Step 5: Interest = amount*years*rate/100.
Step 6: Print interest.
Step 7: Stop.

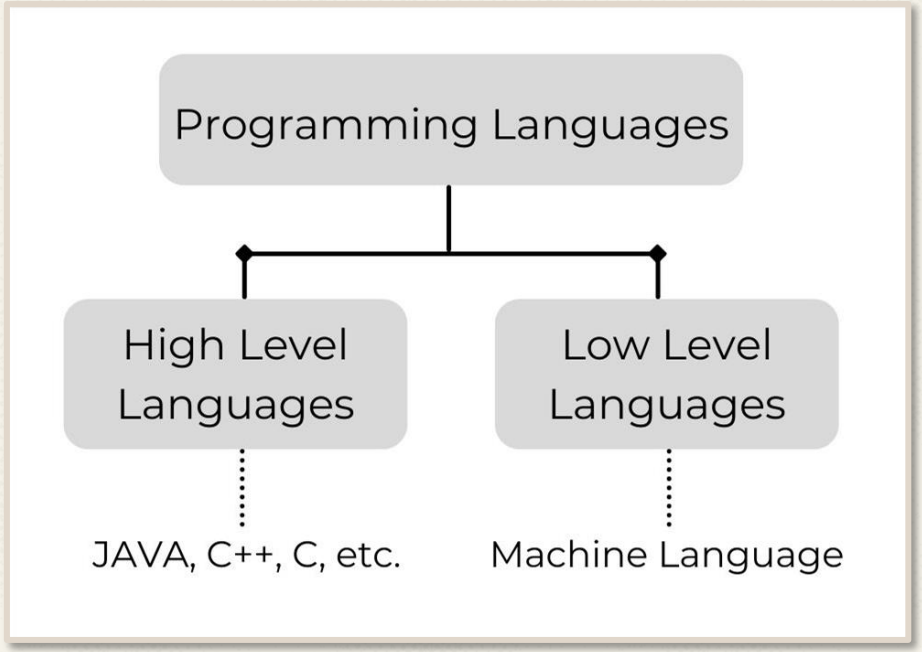
Example



Algorithm

Step 1: Start
Step 2: Input the temperature.
Step 3: Check if it is less than 32.
Step 4: If true, print “Below Freezing”.
Step 5: If false, print “Above Freezing”.
Step 6: Stop.

Programming Languages



Programming Paradigms

Procedural Programming

- This paradigm emphasizes on procedure in terms of underlying machine model.
- It instructs a device how to finish a task in logical steps.

Programming Paradigms

Object Oriented Programming

- Object oriented programming views problems as objects.
- This programming paradigm aims to implement real-world entities like inheritance, hiding and polymorphism in programming.

Getting Started With C++

About C++

- C++ is an object oriented programming language.
- It was developed at AT&T Bell Laboratories, in the early 1980s, by ***Bjarne Stroustrup***.
- Initially called “C with classes”, the name C++ was coined later on by Rick Mascitti.

More about C++

Applications

- Operating Systems
- Banking Applications
- Embedded Systems
- Compilers
- Browsers
- Graphic Software

Character Set

- Letters: A-Z, a-z
- Digits: 0-9
- Special Characters
- White Spaces: Blank space, Horizontal tab, Return, etc.
- Other Characters: 256 ASCII characters

Tokens

Tokens are the smallest individual unit in a program. C++ has the following tokens:

- Keywords
- Identifiers
- Literals
- Punctuators
- Operators

Tokens(contd.)

Keywords

- A word having a special meaning, reserved by the programming language.
- Examples: int, float, do, else, if etc.

Identifiers

- Fundamental building blocks of a program. It is the general terminology used for names of different parts of the program (variables, objects, functions, etc.)

Tokens(contd.)

Rules to follow while constructing Identifiers

1. Names can contain letters, digits and underscores
2. Names must begin with a letter or an underscore (_)
3. Names are case sensitive (myVar and myvar are different variables)
4. Names cannot contain whitespaces or special characters like !, #, %, etc.
5. Reserved words (like C++ keywords, such as int) cannot be used as names

Tokens(contd.)

Literals

- Data items that never change their value during a program.

Punctuators

- [ ] ( ) { } , ; : * = … #

Operators

- Tokens that trigger some computation when applied to variables and other objects in an expression.

Tokens(contd.)

Types of Literals

1. Integer Literal: It is used to represent integer constant.
2. Float Literal: It is used to represent float constant.
3. Character Literal: It is used to represent a single character.
4. String Literal: It is used to represent the character sequence(string).
5. Boolean Literal: It is used to represent Boolean(true or false).

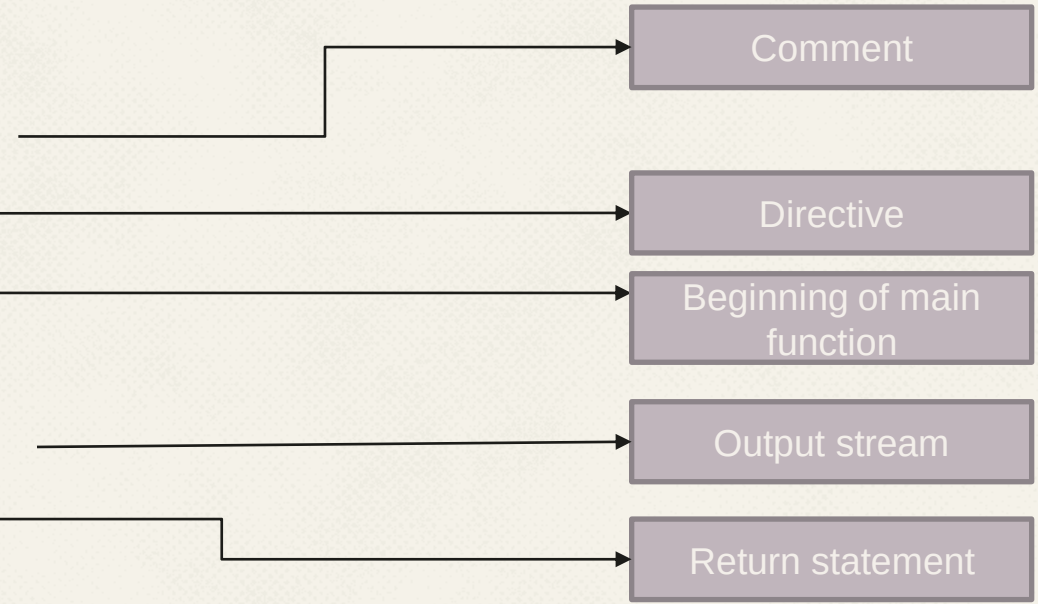
Tokens(contd.)

Escape Sequence

\n	Newline or line feed
\r	Carriage return (Enter)
\t	Horizontal Tab
\v	Vertical Tab
\\	Print back slash
\?	Print question mark
\'	Print single quote
\"	Print double quote
\0	Null character

Structure of a C++ program

```
//My first C++ program
#include<iostream>
int main()
{
    cout<<“Welcome!!”
    return 0;
}
```

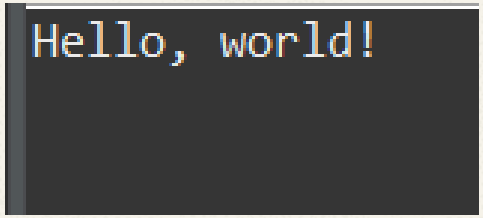


Program 1

```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello, world!" << endl;
  return 0;
}
```

Output:



Data Types

Data types are a means to identify the type of data and the associated operations of handling it.

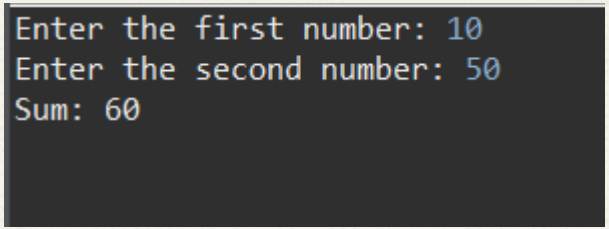
Fundamental Data Types:

a) int
b) char
c) float
d) double
e) void

Program 2

```
#include<iostream>
using namespace std;
int main()
{
      int a, b, sum;
      cout<<“Enter the first number”;
      cin>>a;
 cout<<“Enter the second number”;
      cin>>b;
      sum = a+b;
      cout<<“Sum: ”<<sum;
      return 0;
}
```

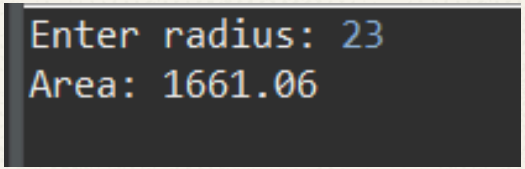
Sample Output:



Program 3

```
#include<iostream>
using namespace std;
int main()
{
      float radius, area;
      cout<<“Enter radius: ”;
      cin>>radius;
      area= 3.14*radius*radius;
      cout<<“Area: ”<<area;
      return 0;
}
```

Sample Output:



Thank You