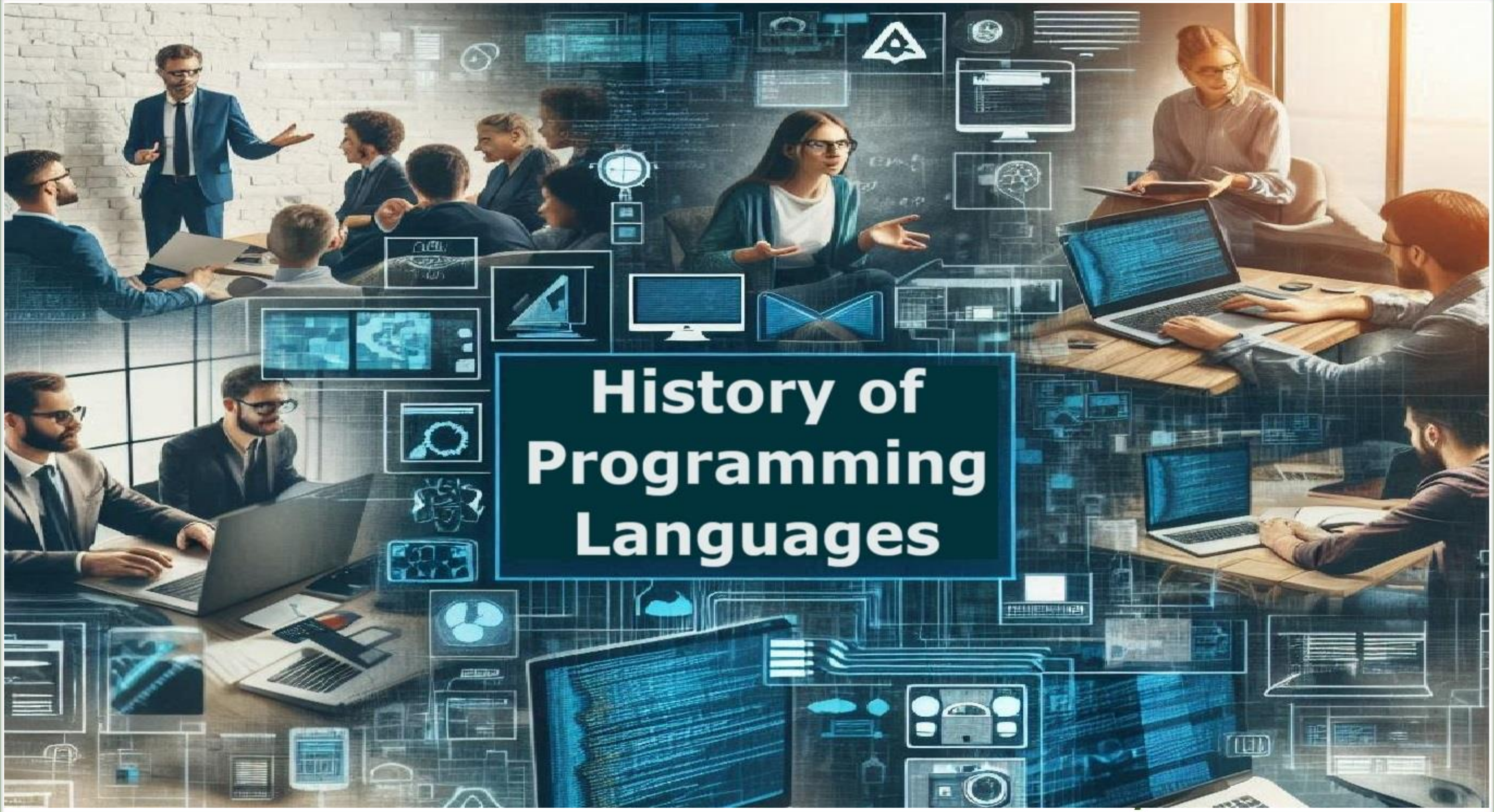


History of Programming Languages

What is Language?

Language is a mode of communication that is used to **share ideas, opinions with each other**. For example, if we want to teach someone, we need a language that is understandable by both communicators.

By – Mohammad Imran

Introduction to Programing Language

Programming languages are used by humans to interact with computers. These languages were invented to have a mode of communication between the two where humans could give instructions for computers to carry out. Over time, several programming languages have developed and today there are about 500+ programming languages. These differ based on their application.

The tools used by software engineers to write down computer packages are programming languages. They are the means of interacting with and commanding computer systems. Numerous distinct programming languages exist, each with its benefits and downsides.

By – Mohammad Imran

Programing Language

Definition

A programming language is a **computer language** that is used by **programmers (developers) to communicate with computers**. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task.

A programming language is mainly used to **develop desktop applications, websites, and mobile applications**.

Programing Language

Need

Programming Language- it is vocabulary and a collection of rules that command a computer, devices, applications to work according to the written codes. The programing language enables us to write efficient programs and develop online solutions such as- mobile applications, web applications, and games, etc.

Programming is used to automate, maintain, assemble, measure and interpret the processing of the data and information. It helps in accelerating the input and output of the devices or applications.

By – Mohammad Imran

Programing Language

Need

Several software packages are made using programming languages, together with:

- ✓ Operating Structures
- ✓ Web Browsers
- ✓ Mobile Applications
- ✓ Desktop Packages
- ✓ Video Games
- ✓ Business - Related Software Programs
- ✓ Embedded Structures

Programing Language

Types

Generally there are two types of programming language.

- ✓ Low Level Programming Language
- ✓ High Level Programming Language

Programing Language

Low Level

Low Level Programming Language is **machine-dependent (0s and 1s)** programming language. The processor runs low- level programs directly without the need of a compiler or interpreter, so the programs written in low-level language can be run very fast.

- ✓ Machine Language
- ✓ Assembly Language

By – Mohammad Imran

Programing Language

High Level

High-level programming language (HLL) is designed for **developing user-friendly software programs and websites**. This programming language requires a compiler or interpreter to translate the program into machine language (execute the program).

The main advantage of a high-level language is that it is **easy to read, write, and maintain**.

Ex: Python, Java, JavaScript, PHP, C#, C++ etc.

By – Mohammad Imran

Programing Language

Middle Level

Middle-level programming language **lies between the low-level programming language and high-level programming language**. It is also known as the intermediate programming language and pseudo-language.

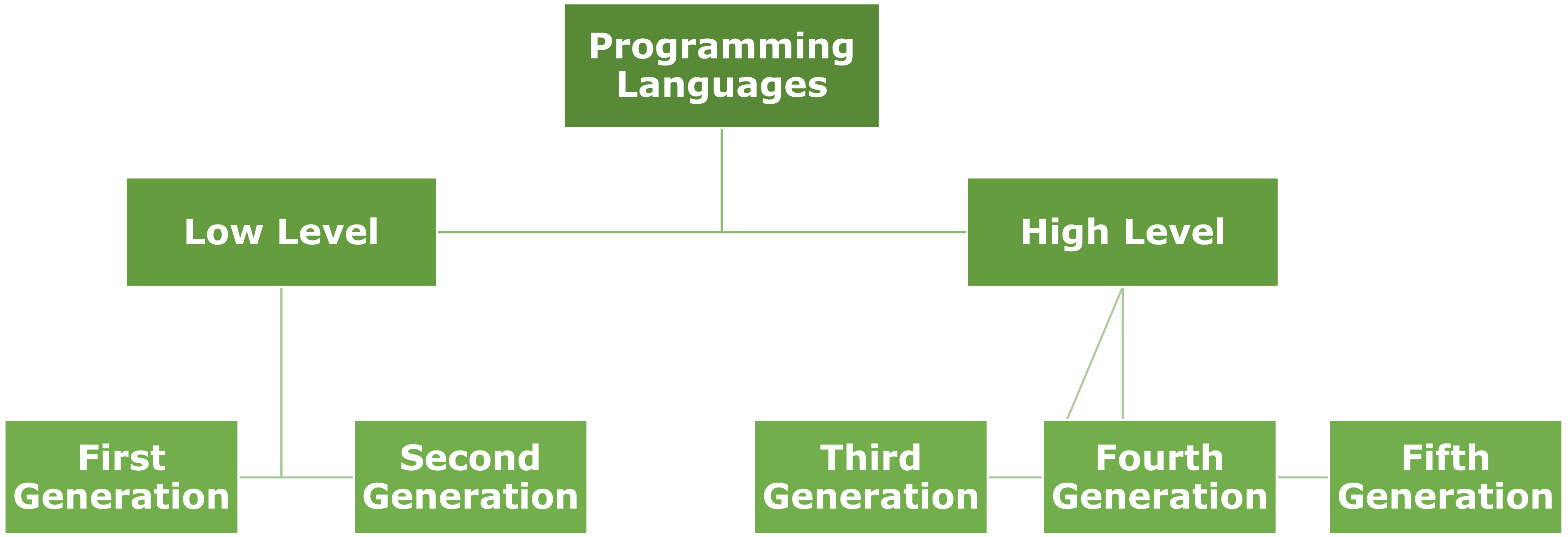
A middle-level programming language's advantages are that it supports the features of high-level programming, it is a user-friendly language, and closely related to machine language and human language.

Example: C, C++, language

By – Mohammad Imran

Programing Language Generation

There are five generations of Programming languages. They are:



By – Mohammad Imran

Programing Language

First Generation

The first-generation languages are also called machine languages / 1G language. This language is machine-dependent. The machine language statements are written in binary code (0/1 form) because the computer can understand only binary language.

Advantages :

- ✓ Fast & efficient as statements are directly written in binary language.
- ✓ No translator is required.

Disadvantages :

- ✓ Difficult to learn binary codes.
- ✓ Difficult to understand – both programs & where the error occurred.

By – Mohammad Imran

Programing Language

Second Generation

The second-generation languages are also called assembler languages / 2G languages. Assembly language contains human-readable notations that can be further converted to machine language using an assembler.

Assembler – Converts assembly level instructions to machine-level instructions.

Programmers can write the code using symbolic instruction codes that are meaningful abbreviations of mnemonics. It is also known as low-level language.

By – Mohammad Imran

Programing Language

Second Generation

Advantages :

- ✓ It is easier to understand if compared to machine language.
- ✓ Modifications are easy.
- ✓ Correction & location of errors are easy.

Disadvantages :

- ✓ Assembler is required.
- ✓ This language is architecture / machine-dependent, with a different instruction set for different machines.

Programing Language

Third Generation

The third generation is also called procedural language / 3GL. It consists of the use of a series of English-like words that humans can understand easily, to write instructions. It's also called High-Level Programming Language. For execution, a program in this language needs to be translated into machine language using a Compiler / Interpreter.

Examples of this type of language are C, PASCAL, FORTRAN, COBOL, etc.

By – Mohammad Imran

Programing Language

Third Generation

Advantages :

- ✓ Use of English-like words makes it a human-understandable language.
- ✓ Lesser number of lines of code as compared to the above 2 languages.
- ✓ Same code can be copied to another machine & executed on that machine by using compiler-specific to that machine.

Disadvantages :

- ✓ Compiler/ interpreter is needed.
- ✓ Different compilers are needed for different machines.

Programing Language

Fourth Generation

The fourth-generation language is also called a non – procedural language / 4GL. It enables users to access the database.

Examples: SQL, Foxpro, Focus, etc.

These languages are also human-friendly to understand.

Fourth Generation

Advantages :

- ✓ Easy to understand & learn.
- ✓ Less time is required for application creation.
- ✓ It is less prone to errors.

Disadvantages :

- ✓ Memory consumption is high.
- ✓ Has poor control over Hardware.
- ✓ Less flexible.

Programing Language

Fifth Generation

The fifth-generation languages are also called 5GL. It is based on the concept of artificial intelligence. It uses the concept that rather than solving a problem algorithmically, an application can be built to solve it based on some constraints, i.e., we make computers learn to solve any problem. Parallel Processing & superconductors are used for this type of language to make real artificial intelligence.

Examples: PROLOG, LISP, etc.

Programing Language Fifth Generation

Advantages :

- ✓ Machines can make decisions.
- ✓ Programmer effort reduces to solve a problem.
- ✓ Easier than 3GL or 4GL to learn and use.

Disadvantages :

- ✓ Complex and long code.
- ✓ More resources are required & they are expensive too.

Complexity

Programming complexity (or **software complexity**) is a term that includes software properties that affect internal interactions. Several commentators distinguish between the terms "complex" and "complicated". Complicated implies being difficult to understand, but ultimately knowable. Complex, by contrast, describes the interactions between entities. As the number of entities increases, the number of interactions between them increases exponentially, making it impossible to know and understand them all. Similarly, higher levels of complexity in software increase the risk of unintentionally interfering with interactions,

Complexity

thus increasing the risk of introducing defects when changing the software. In more extreme cases, it can make modifying the software virtually impossible.

Complexity reflects the number of entities that comprise the software, and the number of interactions between them.

The higher the complexity, the more difficult it is to read the code and maintain it, and the higher the likelihood of faults and defects.

Complexity Type

Problem complexity can be divided into two categories:

- ✓ **Accidental complexity** relates to difficulties a programmer faces due to the software engineering tools. Selecting a better tool set or a higher-level programming language may reduce it. Accidental complexity often results from not using the domain to frame the form of the solution. Domain-driven design can help minimize accidental complexity.
- ✓ **Essential complexity** is caused by the characteristics of the problem to be solved and cannot be reduced.

Security

In computer science, **language-based security (LBS)** refers to a set of techniques used to enhance the security of applications at a high level by leveraging the properties of programming languages. These techniques aim to prevent security vulnerabilities and improve the overall robustness of software systems. By incorporating security constructs directly into the language design, LBS helps detect issues during compilation or execution, reducing the risk of exploitable flaws. If you’re interested in cybersecurity, understanding LBS can be valuable for building secure software.

Features of OOPS

By – Mohammad Imran

Concept of Data Hiding

Data hiding is an object-oriented programming (OOP) technique specifically used to hide internal object details (i.e., data members). Data hiding guarantees exclusive data access to class members only and protects and maintains object integrity by preventing intended or unintended changes and intrusions.

OOPs **Features**

Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

✓ **CLASS**

Class is a blue print of a program. Class has some characteristics and functionality. **Collection of objects** is called class. It is a logical entity. Classes have also **Data Members** (Variables) and **Member Functions**.

For example, let’s say we have a class **Car** which has data members (variables) such as speed, weight, price and functions such as gearChange(), slowDown(), brake() etc.

By – Mohammad Imran

✓ OBJECT

Object is an instance of a class. Any entity that has state and behavior is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.

Class is a user-defined data type and object is a variable of class type. Object is used to access class members.
Object is a runtime entity, it is created at runtime.

✓ INHERITANCE

When one object acquires all the properties and behaviors of parent object i.e. known as inheritance. It provides code reusability.

Inheritance means access the properties and features of one class into another class. The class who is going to provide its features to another class will be called base class and the class who is using the properties and features of another class will be called derived class.

✓ POLYMORPHISM

Polymorphism is taken by Greek work **‘Poly’** means **‘Many’** and **‘Morphos’** means ‘**Form’** it means **many forms**.

When **one task is performed by different ways** known as polymorphism. For example: to convince the customer differently.

In C++, we use Function overloading and Function overriding to achieve polymorphism.

✓ ENCAPSULATION

Binding (or wrapping) code and data together into a single unit is known as encapsulation. For example: capsule, it is wrapped with different medicines.

Encapsulation is a process of wrapping data members and member functions in a single unit called class. Using the method of encapsulation, the programmer cannot directly access the data. Data is only accessible through the object of the class.

✓ ABSTRACTION

Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.

The basic idea of data abstraction is, visible only the necessary information, unnecessary information will be hidden from the outside world. This can be done by making class members as private members of class. Private members can be accessed only within the same class where they are declared.

Introduction To C++

By – Mohammad Imran

MyAnatomy

Introduction

History of C++ language is interesting to know. Here we are going to discuss brief history of C++ language.

It was develop for adding a feature of **OOP (Object Oriented Programming)** in C without significantly changing the C component.

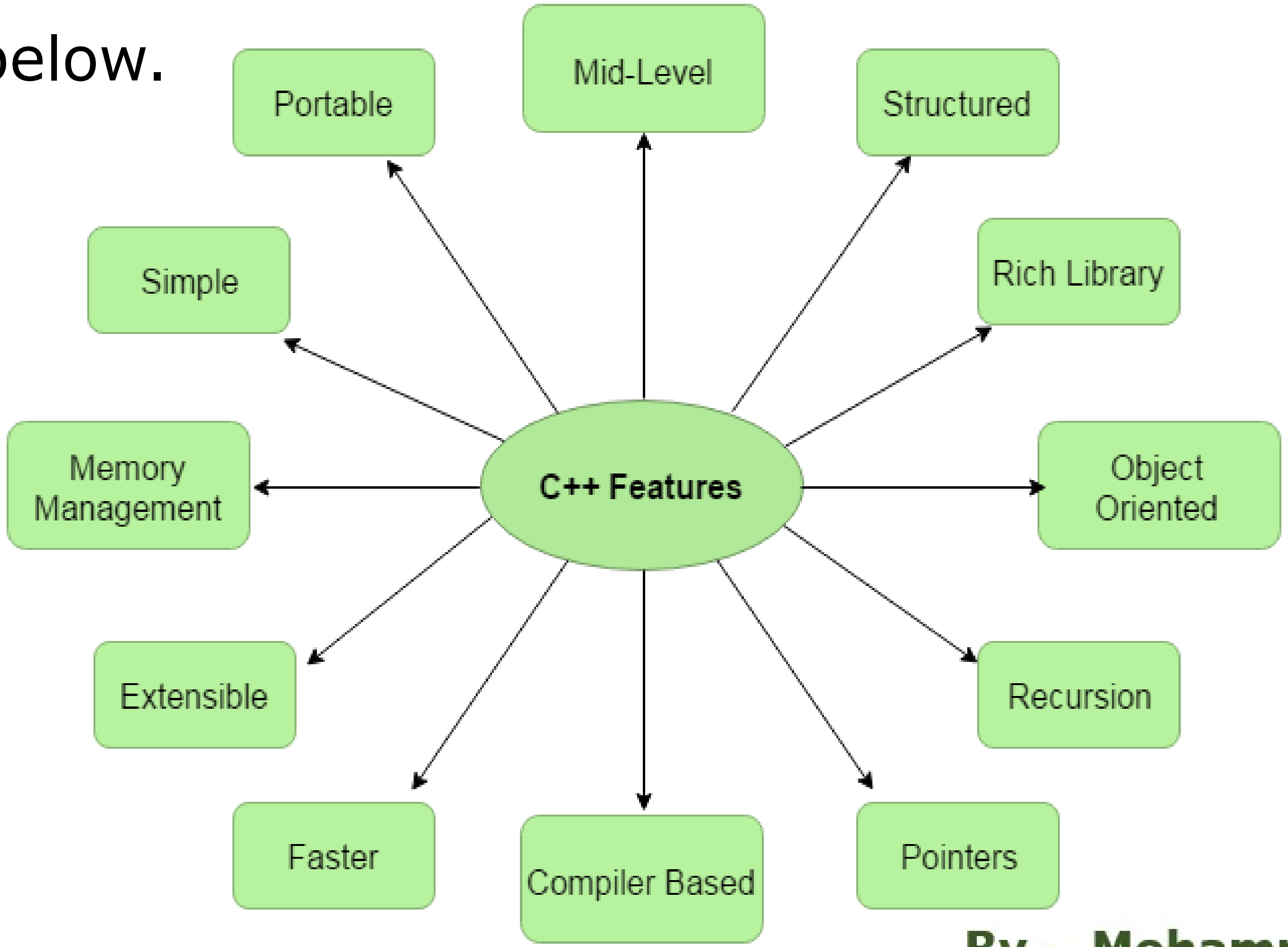
C++ programming is "relative" (called a superset) of C, it means any valid C program is also a valid C++ program.

By – Mohammad Imran

C++ Features

MyAnatomy

C++ is object oriented programming language. It provides a lot of features that are given below.



By – Mohammad Imran

Features

MyAnatomy

- ✓ **Simple**

  C++ is a simple language in the sense that it provides structured approach (to break the problem into parts), rich set of library functions, data types etc.

- ✓ **Machine Independent or Portable**

  Unlike assembly language, c programs can be executed in many machines with little bit or no change. But it is not platform-independent.

By – Mohammad Imran

C++ **Features**

MyAnatomy

✓ Mid-Level Programming Language

C++ is also used to do low level programming. It is used to develop system applications such as kernel, driver etc. It also supports the feature of high level language. That is why it is known as mid-level language.

✓ Structured Programming Language

C++ is a structured programming language in the sense that we can break the program into parts using functions. So, it is easy to understand and modify.

By – Mohammad Imran

MyAnatomy

- ✓ **Rich Library**

  C++ provides a lot of inbuilt functions that makes the development fast.

- ✓ **Memory Management**

  It supports the feature of dynamic memory allocation. In C++ language, we can free the allocated memory at any time by calling the **free()** function.

- ✓ **Speed**

  The compilation and execution time of C++ language is fast.

By – Mohammad Imran

Features

MyAnatomy

✓ Pointer

C++ provides the feature of pointers. We can directly interact with the memory by using the pointers. We can use pointers for memory, structures, functions, array etc.

✓ Recursion

In C++, we can call the function within the function. It provides code reusability for every function.

✓ Extensible

C++ language is extensible because it can easily adopt new features.

C++ Features

- ✓ **Object-Oriented**

  C++ is object oriented programming language. OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.

- ✓ **Compiler Based**

  C++ is a compiler based programming language, it means without compilation no C++ program can be executed. First we need to compile our program using compiler and then we can execute our program.

Following features of C++ gives more advantages over C Language

- ✓ There is Stronger Type Checking in C++.
- ✓ All the OOPS features in C++ like Abstraction, Encapsulation, Inheritance etc makes it more worthy and useful for programmers.
- ✓ C++ supports and allows user defined operators (i.e Operator Overloading) and function overloading is also supported.

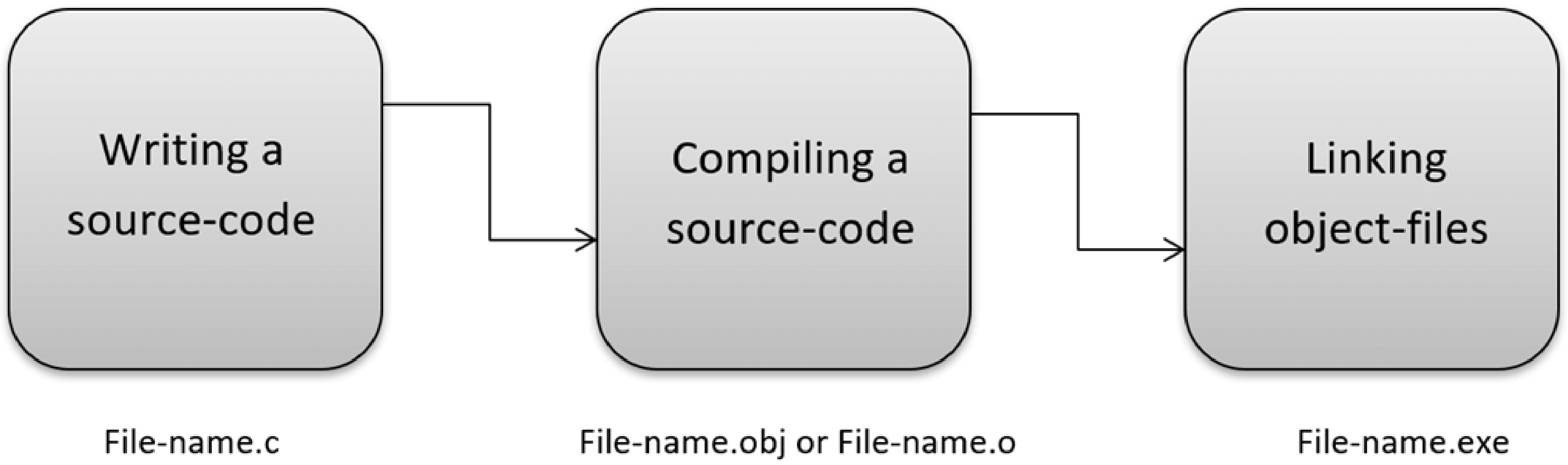
OOPs **Features**

- ✓ Exception Handling is there in C++.
- ✓ The Concept of Virtual functions and also Constructors and Destructors for Objects.
- ✓ Inline Functions in C++ instead of Macros in C language. Inline functions make complete function body act like Macro, safely.
- ✓ Variables can be declared anywhere in the program in C++, but must be declared before they are used.

OOPs C++

How it Works?

C++ is a compiled language. A compiler is a special tool that compiles the program and converts it into the object file which is machine readable. After the compilation process, the linker will combine different object files and creates a single executable file to run the program. The following diagram shows the execution of a 'C++' program.



By – Mohammad Imran

First Program

C++ is a compiled language. A compiler is a special tool that compiles the program and converts it into the object file which is machine readable. After the compilation process, the linker will combine different object files and creates a single executable file to run the program. The following diagram shows the execution of a 'C++' program.