1

Importance of C & C++

Objectives

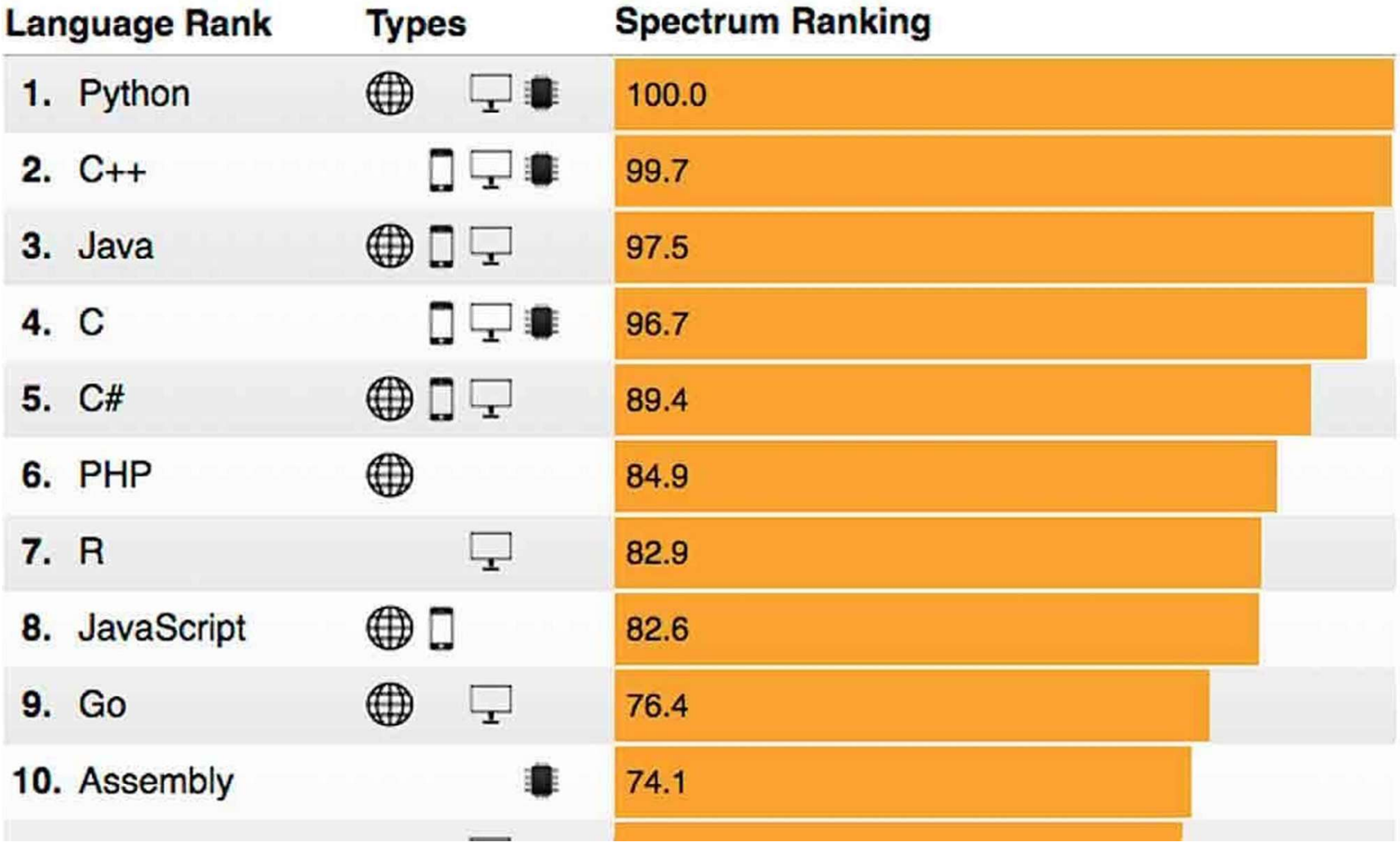
After completing this lesson, you should be able to do the following:

- What is C C++ ?
- Why do we need C++ ?
- Basic Concepts of C++
- Differences Between C & C++

Introduction to C C++

What is C Programming Langauge?

- **C** is a general-purpose programming language that is extremely popular, simple, and flexible to use.
- It is a structured programming language that is machine-independent and extensively used to write various applications, Operating Systems like Windows, and many other complex programs like Oracle database, Git, Python interpreter, and more.
- It is said that ***‘C’ is a god’s programming language***. One can say, C is a base for the programming. If you know ‘C,’ you can easily grasp the knowledge of the other programming languages that uses the concept of ‘C’



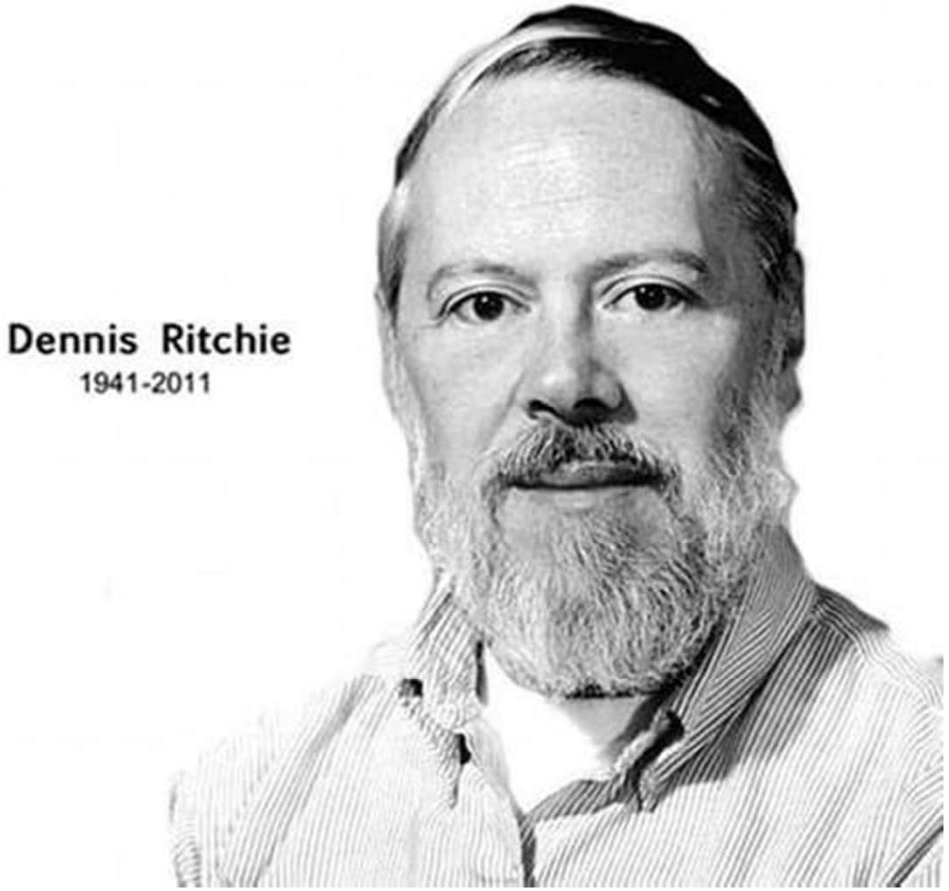
Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥 ▣	100.0
2. C++	📱 🖥 ▣	99.7
3. Java	🌐 📱 🖥	97.5
4. C	📱 🖥 ▣	96.7
5. C#	🌐 📱 🖥	89.4
6. PHP	🌐	84.9
7. R	🖥	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥	76.4
10. Assembly	▣	74.1

History of C language

- The base or father of programming languages is ‘ALGOL.’ It was first introduced in 1960. ‘ALGOL’ was used on a large basis in European countries.
- ‘ALGOL’ introduced the concept of structured programming to the developer community.
- In 1967, a new computer programming language was announced called as ‘BCPL’ which stands for Basic Combined Programming Language.
- BCPL was designed and developed by Martin Richards, especially for writing system software.

- This was the era of programming languages. Just after three years, in 1970 a new programming language called ‘B’ was introduced by Ken Thompson that contained multiple features of ‘BCPL.’
- This programming language was created using UNIX operating system at AT&T and Bell Laboratories. Both the ‘BCPL’ and ‘B’ were system programming languages.
- In 1972, a great computer scientist Dennis Ritchie created a new programming language called ‘C’ at the Bell Laboratories. It was created from ‘ALGOL’, ‘BCPL’ and ‘B’ programming languages.
- ‘C’ programming language contains all the features of these languages and many more additional concepts that make it unique from other languages.

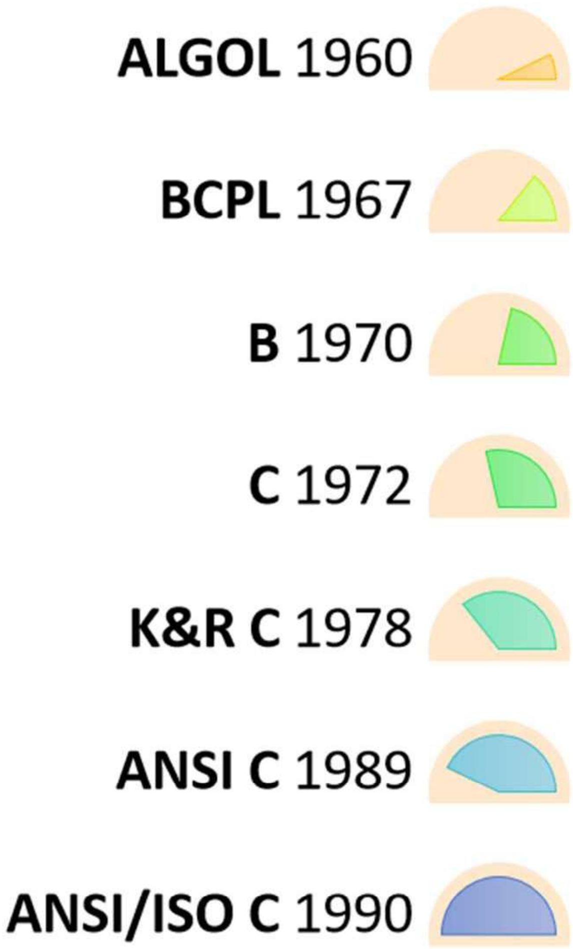
Father of C Programming



Dennis Ritchie
1941-2011

- ‘C’ is a powerful programming language which is strongly associated with the UNIX operating system.
- Even most of the UNIX operating system is coded in ‘C’. Initially ‘C’ programming was limited to the UNIX operating system, but as it started spreading around the world, it became commercial, and many compilers were released for cross-platform systems.
- Today ‘C’ runs under a variety of operating systems and hardware platforms. As it started evolving many different versions of the language were released.
- At times it became difficult for the developers to keep up with the latest version as the systems were running under the older versions.

- To assure that ‘C’ language will remain standard, American National Standards Institute (ANSI) defined a commercial standard for ‘C’ language in 1989.
- Later, it was approved by the International Standards Organization (ISO) in 1990. ‘C’ programming language is also called as ***‘ANSI C’.***



ALGOL 1960

BCPL 1967

B 1970

C 1972

K&R C 1978

ANSI C 1989

ANSI/ISO C 1990

- Languages such as C++/Java are developed from ‘C’. These languages are widely used in various technologies.
- Thus, ‘C’ forms a base for many other languages that are currently in use.

1. C as a mother language

- C language is considered as the mother language of all the modern programming languages because **most of the compilers, JVMs, Kernels, etc. are written in C language**, and most of the programming languages follow C syntax, for example, C++, Java, C#, etc.
- It provides the core concepts like the array, strings, functions, file handling, etc. that are being used in many languages like C++, Java, C#, etc.

2. C as a system programming language

- A system programming language is used to create system software. C language is a system programming language because it **can be used to do low-level programming (for example driver and kernel)**. It is generally used to create hardware devices, OS, drivers, kernels, etc. For example, Linux kernel is written in C.
- It can't be used for internet programming like Java, .Net, PHP, etc.

3. C as a procedural language

- A procedure is known as a function, method, routine, subroutine, etc. A procedural language **specifies a series of steps for the program to solve the problem**.
- A procedural language breaks the program into functions, data structures, etc.
- C is a procedural language. In C, variables and function prototypes must be declared before being used.

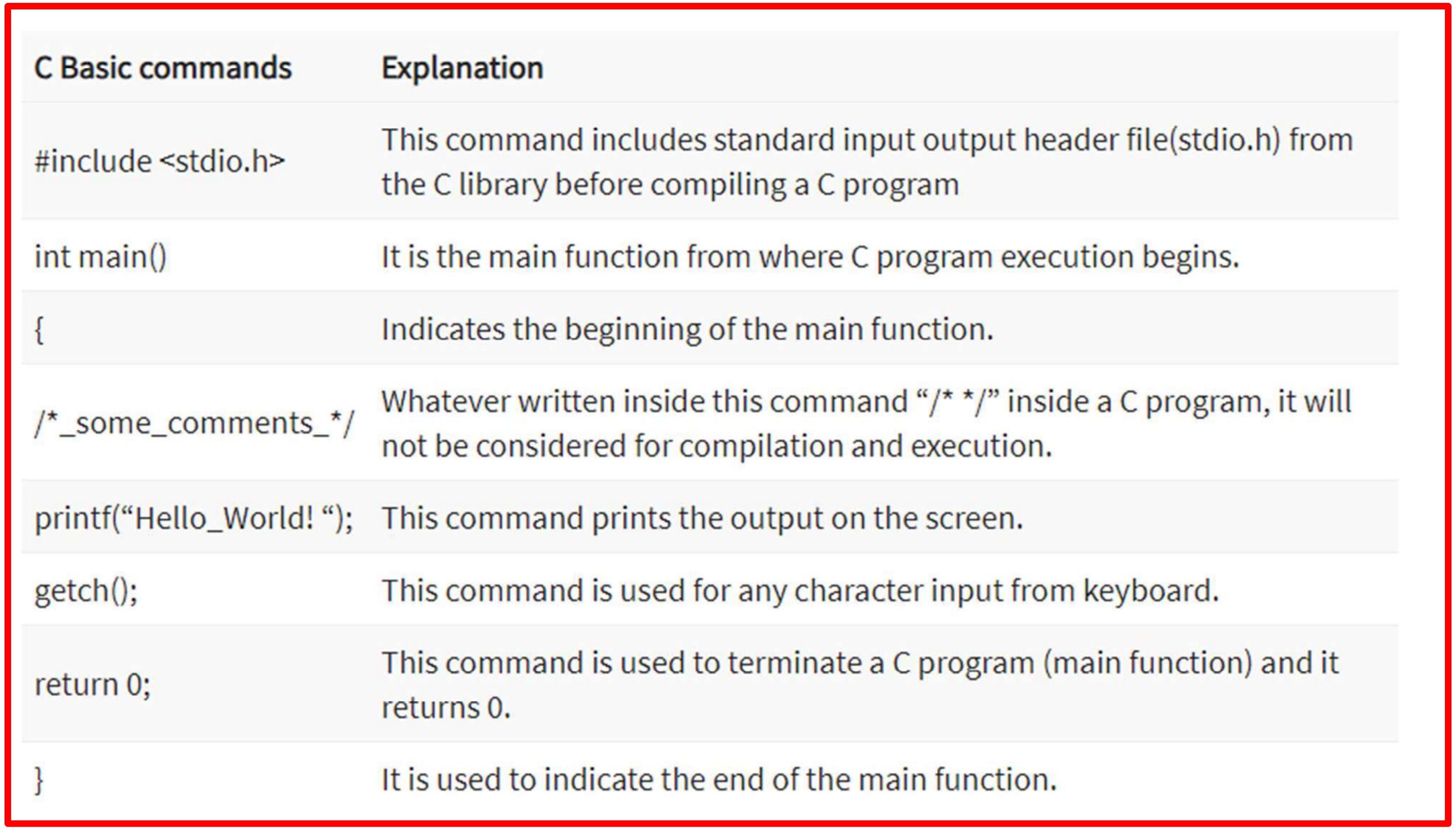
4. C as a structured programming language

- A structured programming language is a subset of the procedural language. **Structure means to break a program into parts or blocks** so that it may be easy to understand.
- In the C language, we break the program into parts using functions. It makes the program easier to understand and modify.

5. C as a mid-level programming language

- C is considered as a middle-level language because it **supports the feature of both low-level and high-level languages**. C language program is converted into assembly code, it supports pointer arithmetic (low-level), but it is machine independent (a feature of high-level).
- A **Low-level language** is specific to one machine, i.e., machine dependent. It is machine dependent, fast to run. But it is not easy to understand.
- A **High-Level language** is not specific to one machine, i.e., machine independent. It is easy to understand.

C Basic Commands



C Basic commands	Explanation
#include <stdio.h>	This command includes standard input output header file(stdio.h) from the C library before compiling a C program
int main()	It is the main function from where C program execution begins.
{	Indicates the beginning of the main function.
/*_some_comments_*/	Whatever written inside this command “/* */” inside a C program, it will not be considered for compilation and execution.
printf(“Hello_World! “);	This command prints the output on the screen.
getch();	This command is used for any character input from keyboard.
return 0;	This command is used to terminate a C program (main function) and it returns 0.
}	It is used to indicate the end of the main function.

Example

```
// Sample HelloWorld code
#include <stdio.h>
int main()  {
        printf("Hello C Programming\n");
        return 0;
}
```

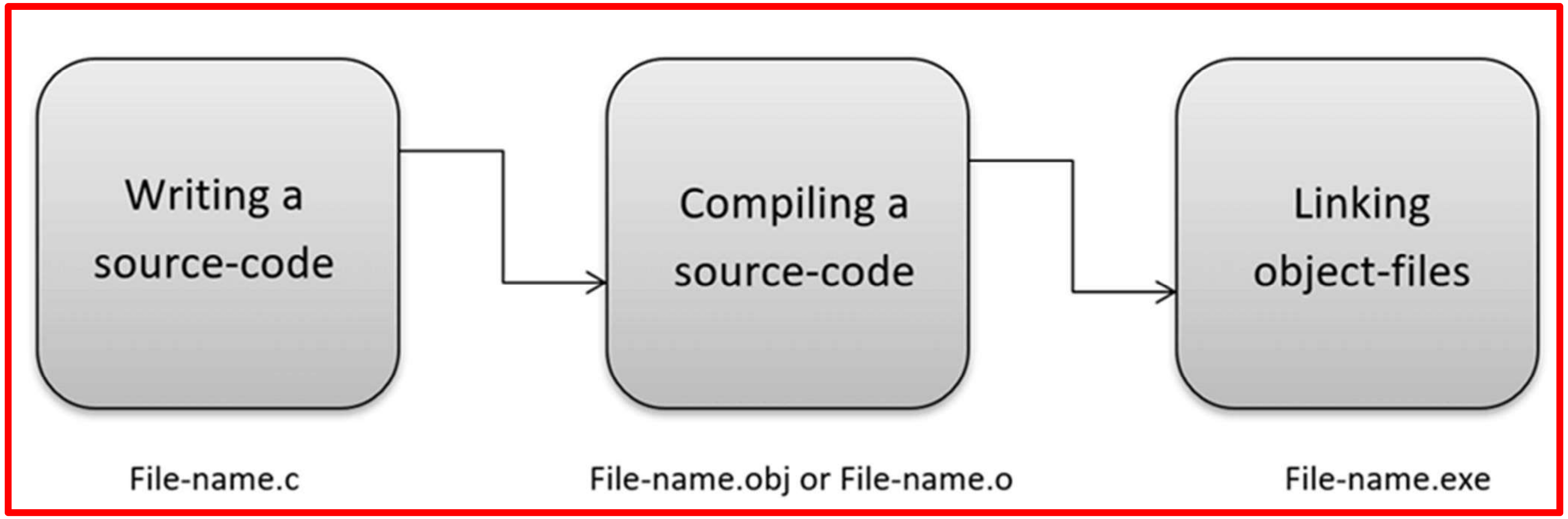
Where is C used? Key Applications

1. ‘C’ language is widely used in embedded systems.
2. It is used for developing system applications.
3. It is widely used for developing desktop applications.
4. Most of the applications by Adobe are developed using ‘C’ programming language.
5. It is used for developing browsers and their extensions. Google’s Chromium is built using ‘C’ programming language.
6. It is used to develop databases. MySQL is the most popular database software which is built using ‘C’.

7. It is used in developing an operating system. Operating systems such as Apple’s OS X, Microsoft’s Windows, and Symbian are developed using ‘C’ language. It is used for developing desktop as well as mobile phone’s operating system.
8. It is used for compiler production.
9. It is widely used in IOT applications.

How C Programming Language 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



Features of C Language

C is the widely used language. It provides many **features** that are given below.

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

2. Machine Independent or Portable
   - Unlike assembly language, c programs **can be executed on different machines** with some machine specific changes. Therefore, C is a machine independent language.

3. Mid-level programming language
    - Although, C is **intended to do low-level programming**. It is used to develop system applications such as kernel, driver, etc. It **also supports the features of a high-level language**. That is why it is known as mid-level language.

4. 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. Functions also provide code reusability.

5. Rich Library
    - C **provides a lot of inbuilt functions** that make the development fast.

6. 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.
7. Speed
   - The compilation and execution time of C language is fast since there are lesser inbuilt functions and hence the lesser overhead.
8. 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.

9) Recursion

- In C, we **can call the function within the function**. It provides code reusability for every function. Recursion enables us to use the approach of backtracking.

10) Extensible

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

Applications of C Language

- C was used in programs that were used in making operating systems. C was known as a system development language because the code written in C runs as fast as the code written in assembly language.
  - *Operating Systems*
  - *Language Compilers*
  - *Assemblers*
  - *Text Editors*
  - *Print Spoolers*
  - *Network Drivers*
  - *Modern Programs*
  - *Databases*
  - *Language Interpreters*
  - *Utilities*

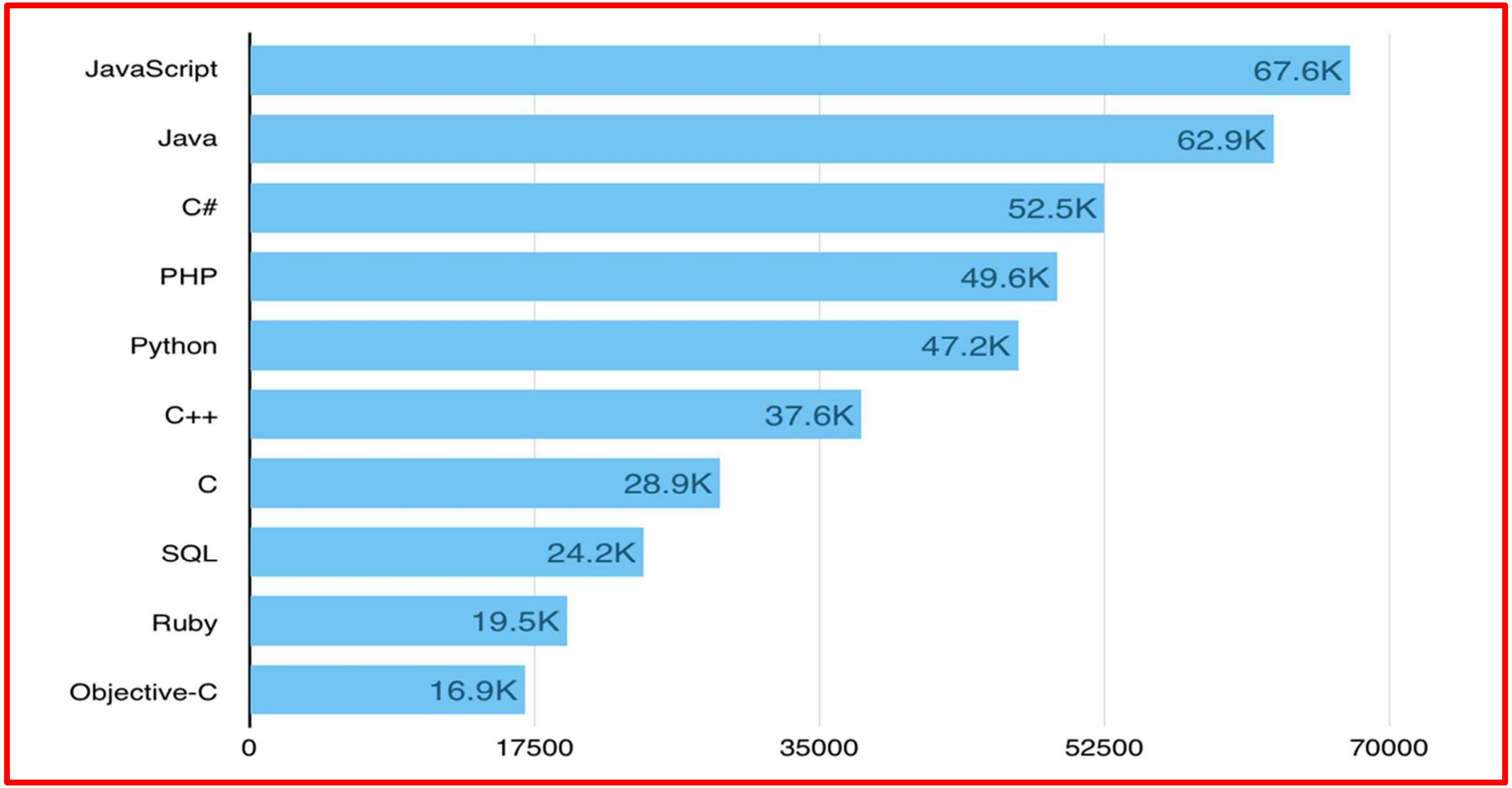
Disadvantages of C Programming

1. Does not support Object-Oriented Programming (OOP) features such as Polymorphism, Inheritance, and Encapsulation.
2. The bugs are not detected after each line of code, instead it is detected after writing the entire program. This makes debugging in C difficult for complex programs.
3. C does not support the concept of namespace, due to which two variables cannot be declared with same name.
4. C does not Exception Handling which in other languages are used to detect the bugs and generate appropriate responses.
5. C language has low level of abstraction. Due to this C has minimum data hiding and this affects the overall security of the language.

Introduction to C++

What is C++?

- C++ is a general-purpose, object-oriented programming language. It was created by Bjarne Stroustrup at Bell Labs circa 1980.
- C++ is very similar to C (invented by Dennis Ritchie in the early 1970s). C++ is so compatible with C that it will probably compile over 99% of C programs without changing a line of source code.
- Though C++ is a lot of well-structured and safer language than C as it OOPs based.
- Some computer languages are written for a specific purpose.
  - Like, Java was initially devised to control toasters and some other electronics.
  - C was developed for programming OS.
  - Pascal was conceptualized to teach proper programming techniques.
  - But C++ is a general-purpose language. It well deserves the widely acknowledged nickname “Swiss Pocket Knife of Languages.”


JavaScript
Java
C#
PHP
Python
C++
C
SQL
Ruby
Objective-C
67.6K
62.9K
52.5K
49.6K
47.2K
37.6K
28.9K
24.2K
19.5K
16.9K
0
17500
35000
52500
70000

Is C++ best programming language?

- The answer depends on perspective and requirements. Some tasks can be done in C++, though not very quickly. For example, designing GUI screens for applications.
- Other languages like Visual Basic, Python have GUI design elements built into them. Therefore, they are better suited for GUI type of task.
- Some of the scripting languages that provide extra programmability to applications. Such as MS Word and even photoshop tend to be variants of Basic, not C++.
- C++ is still used widely, and the most famous software have their backbone in C++.

Who uses C++?

Some of today's most visible used systems have their critical parts written in C++.

Examples are

- Amadeus (airline ticketing)
- Bloomberg (financial formation),
- Amazon (Web commerce), Google (Web search)
- Facebook (social media)

Cont …

Many programming languages depend on C++’s performance and reliability in their implementation. Examples include:

- Java Virtual Machines
- JavaScript interpreters (e.g., Google’s V8)
- Browsers (e.g., Internet Explorer, Mozilla’s Firefox, Apple’s Safari, and Google’s Chrome)
- Application and Web frameworks (e.g., Microsoft’s .NET Web services framework).
- Applications that involve local and wide area networks, user interaction, numeric, graphics, and database access highly depend on C++ language.

Five Basic Concepts of C++

C++ Variables

- Variables are the backbone of any programming language.
- A variable is merely a way to store some information for later use. We can retrieve this value or data by referring to a “word” that will describe this information.
- Once declared and defined they may be used many times within the scope in which they were declared.

C++ Control Structures

- When a program runs, the code is read by the compiler line by line (from top to bottom, and for the most part left to right). This is known as “**code flow.**“
- When the code is being read from top to bottom, it may encounter a point where it **needs to make a decision**. Based on the decision, the program may jump to a different part of the code. It may even make the compiler re-run a specific piece again, or just skip a bunch of code.

C++ Data Structures

- A **data structure** is a great **way to get around having to create thousands of variables.** C++ contains many types of inbuilt data structures.

C++ Syntax

- The syntax is a layout of words, expression, and symbols.
- Well, it’s because an email address has its well-defined syntax. You need some combination of letters, numbers, potentially with underscores (_) or periods (.) in between, followed by an at the rate (@) symbol, followed by some website domain (company.com).
- So, syntax in a programming language is much the same. They are some well-defined set of rules that allow you to create some piece of well-functioning software.
- But, if you don’t abide by the rules of a programming language or syntax, you’ll get errors.

C++ Tools

- In the real world, a tool is something (usually a physical object) that helps you to get a certain job done promptly.
- Well, this holds true with the programming world too. A tool in programming is some piece of software which when used with the code allows you to program faster.
- Most crucial tool, considered by many, is an IDE, an **Integrated Development Environment.** An IDE is a software which will make your coding life so much easier. IDEs ensure that your files and folders are organized and give you a nice and clean way to view them.

Use of C++ Programming Language

- **Operating Systems**
  - Whether it is Microsoft Windows or Mac OSX or Linux – all of the operating systems have some parts which are programmed in C++. It is the backbone of all the well-known OSs as C++ is a strongly typed and quick programming language, that makes it an ideal choice for developing an operating system.
- **Games**
  - Because of the fact that it is one of the fastest programming languages, C++ is widely used in programming of game development engines. C++ can easily manipulate hardware resources and it can also provide procedural programming for CPU intensive functions.

- **Browsers**
  - The rendering engines of various web browsers are programmed in C++ because of the speed it offers.
- **Libraries**
  - Many high-level libraries use C++ as the core programming language. For example, several Machine Learning libraries use C++ in the backend because of its speed.
- **Graphics**
  - C++ is widely used in almost all graphics applications that require fast rendering, image processing, real-time physics and mobile sensors.

- **Banking Applications**
  - One of the most popularly used core-banking systems – Infosys Finacle, uses C++ as the backend programming language. Banking applications need to process millions of transactions on a daily basis and require high concurrency and low latency support.
- **Cloud/Distributed Systems**
  - Cloud storage systems use scalable file-systems that work close to the hardware. That's why C++ becomes a preferred choice for Cloud systems.

- **Embedded Systems**
  - Various embedded systems like medical machines, smartwatches, etc., use C++ as the primary programming language.
- **Compilers**
  - Compilers of various programming languages use C++ as the backend programming language.

Features of C++

1. **Object-oriented programming:** C++ supports object-oriented programming (OOP) concepts such as inheritance, polymorphism, and encapsulation, which makes it easier to develop large, complex software systems.
2. **Templates:** C++ supports templates, which are reusable code blocks that can be used to create generic functions and data types.
3. **Exception handling:** C++ has built-in support for exception handling, which allows you to catch and handle errors and exceptions that occur during program execution.
4. **Function overloading:** C++ allows you to create multiple functions with the same name but with different parameters, a technique known as function overloading.

5. **Operator overloading:** C++ allows you to overload operators, which means that you can define how operators behave when used with different data types.
6. **Namespaces:** C++ supports namespaces, which are used to organize code and prevent name collisions.
7. **Standard Template Library (STL):** C++ includes the Standard Template Library (STL), which is a collection of templates and algorithms that can be used to implement common data structures and algorithms.
8. **Compatibility with C:** C++ is fully compatible with C, which means that you can use C libraries and code in your C++ programs. This makes it easy to incorporate existing C code into C++ projects.

Example

```
#include <iostream>

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

Example

```
#include <iostream>

class Greeter {
public:
    void sayHello() const {
        std::cout << "Hello, world!" << std::endl;
    }
};

int main() {
    Greeter greeter;
    greeter.sayHello();
    return 0;
}
```

Example

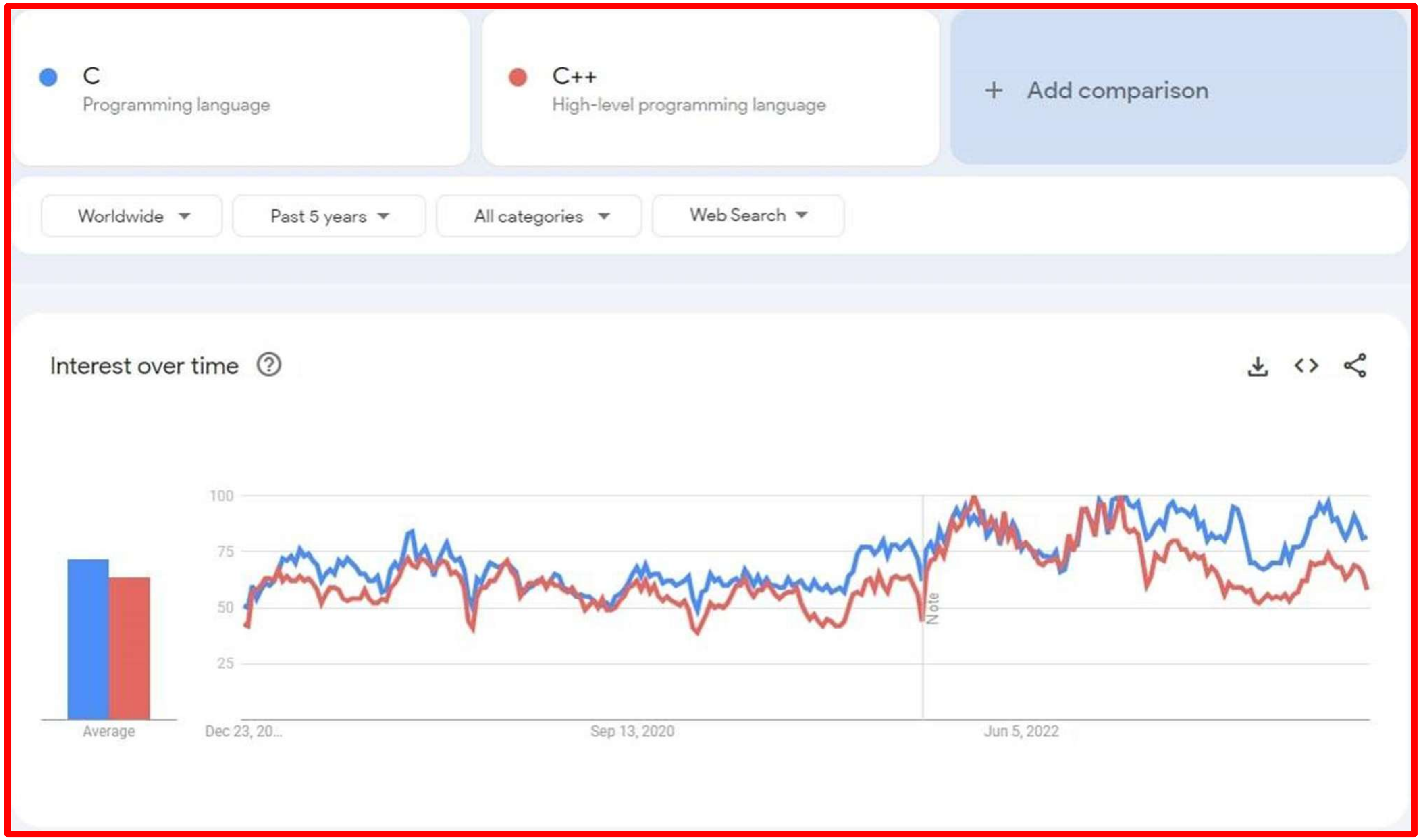
```
// Driver code
    #include <iostream>
class Student {
   public:
      int id;//data member (also instance variable)
      string name;//data member(also instance variable)
};
int main() {
    Student s1; //creating an object of Student
    s1.id = 201;
    s1.name = “Rahul Dravid";
    cout<<s1.id<<endl;
    cout<<s1.name<<endl;
    return 0;
}
```

Disadvantages of C++

1. For a beginner, C++ might be a relatively difficult language to learn and start with.
2. Manual memory management in C++ is a hassle, as there is no garbage collection to automatically remove unwanted data.
3. If you're used to automated memory management, managing memory allocation manually can be a challenge.
4. Not very secure because of pointers, global variables, etc.
5. C++ does not support built-in code threads due to which the process is slower and complicated.
6. The pointers in C++ take up more than the required memory which might not be suitable for some devices.

Similarities Between C & C++

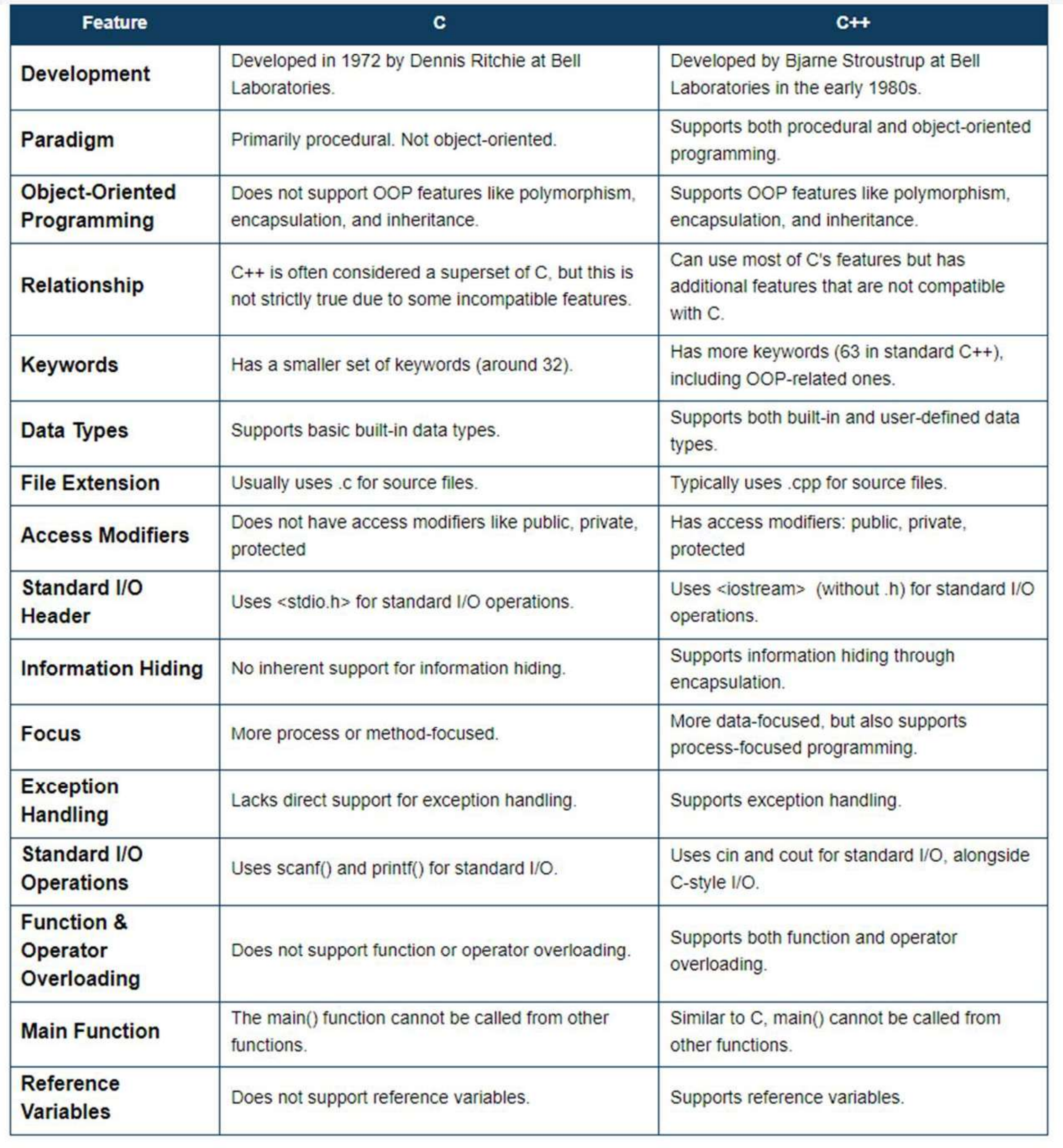
Google Trends Comparison - C vs C++



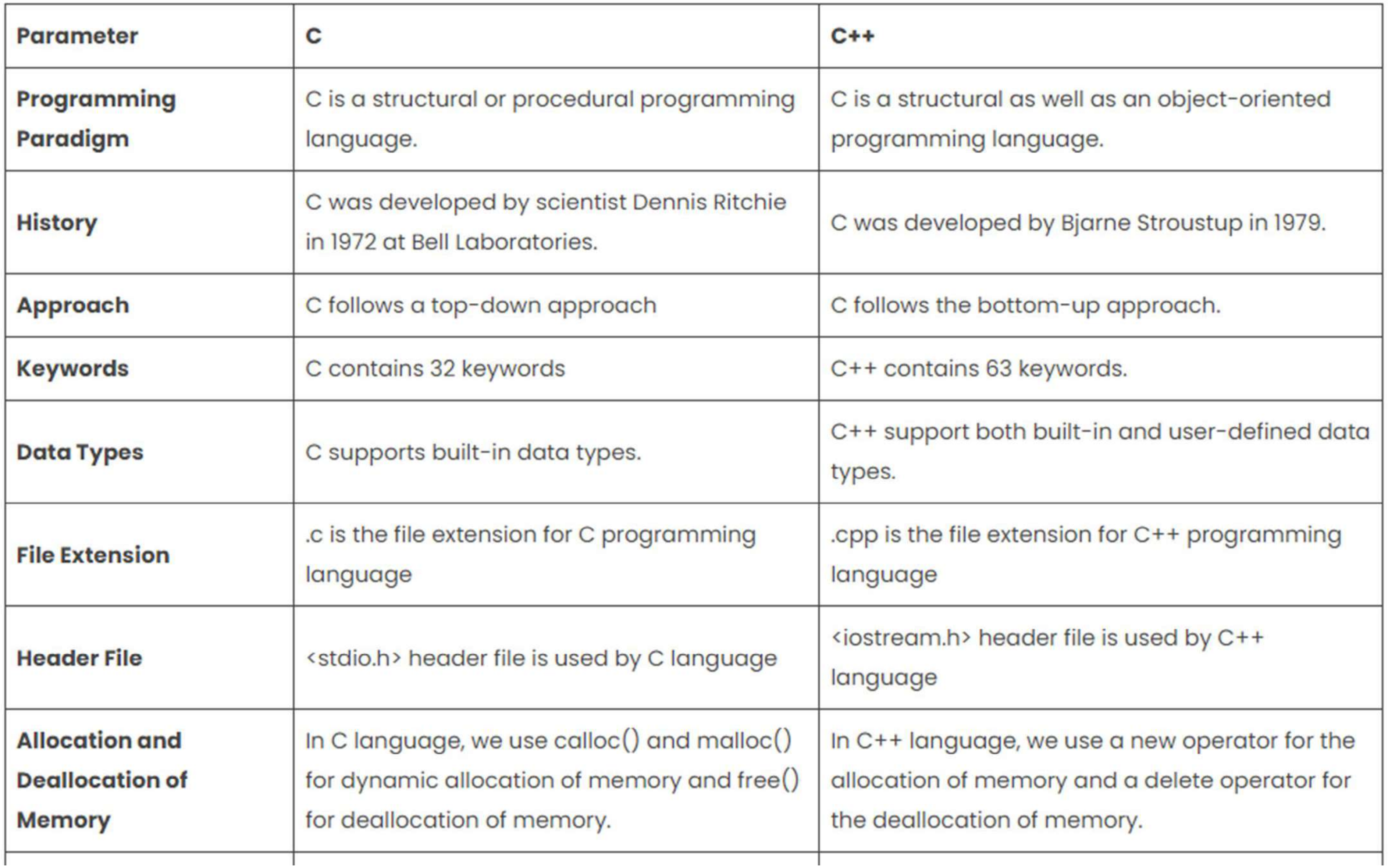
Similarities Between C & C++

1. Both the languages have a similar syntax.
2. Code structure of both the languages are same.
3. The compilation of both the languages is similar.
4. They share the same basic syntax. Nearly all of C's operators and keywords are also present in C++ and do the same thing.
5. C++ has a slightly extended grammar than C, but the basic grammar is the same.
6. Basic memory model of both is very close to the hardware.
7. Same notions of stack, heap, file-scope and static variables are present in both the languages.

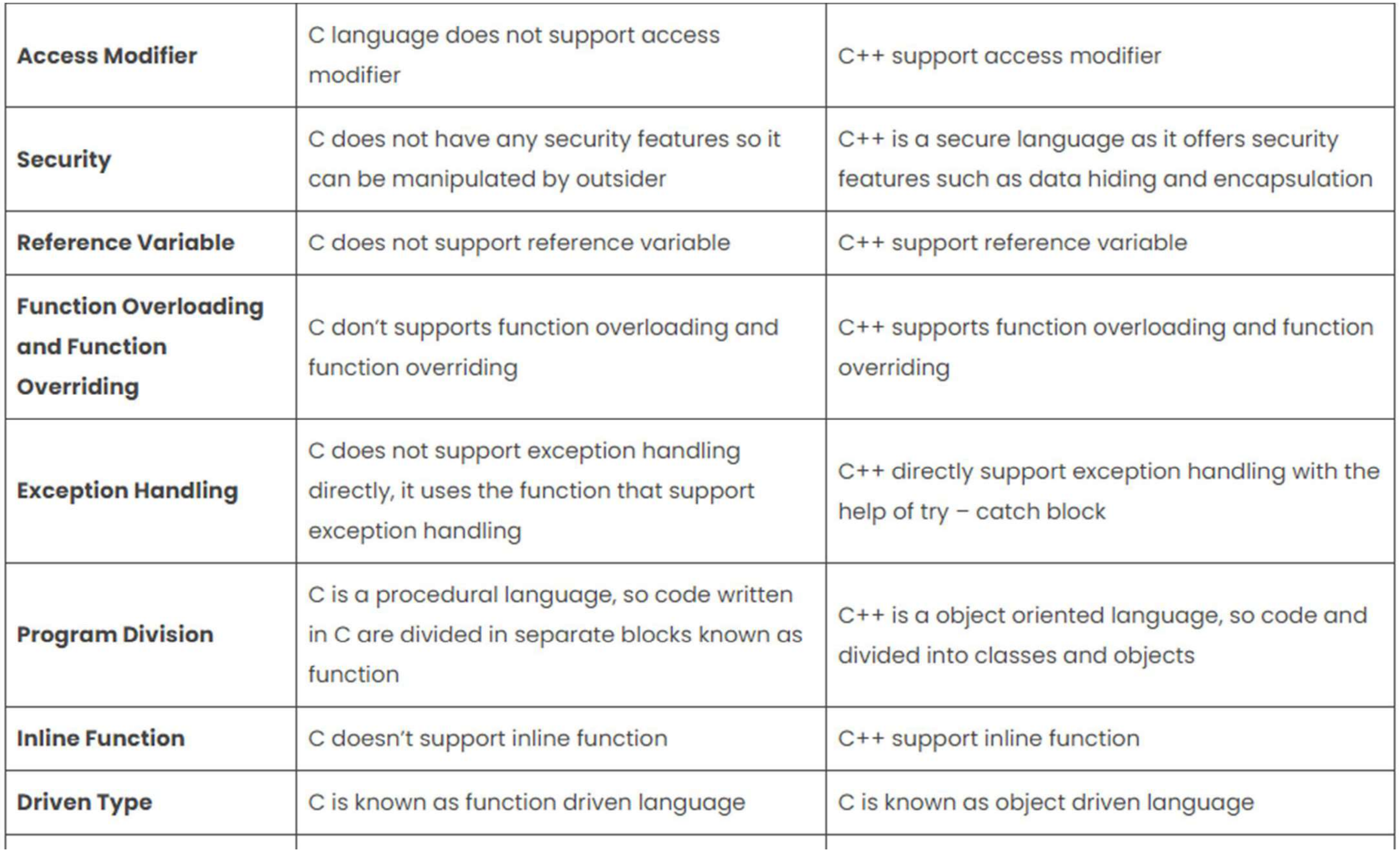
Differences Between C & C++



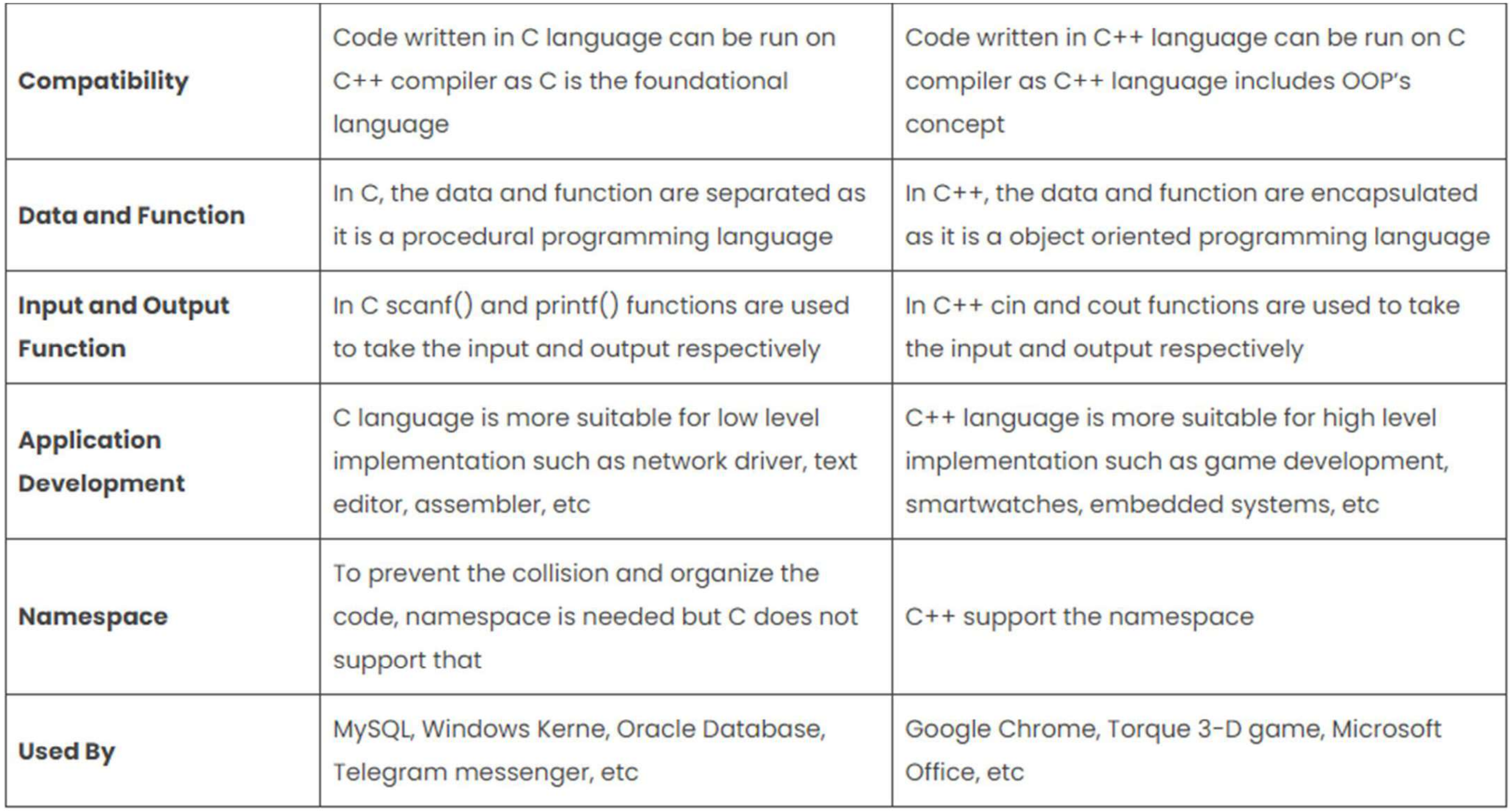
Feature	C	C++
Development	Developed in 1972 by Dennis Ritchie at Bell Laboratories.	Developed by Bjarne Stroustrup at Bell Laboratories in the early 1980s.
Paradigm	Primarily procedural. Not object-oriented.	Supports both procedural and object-oriented programming.
Object-Oriented Programming	Does not support OOP features like polymorphism, encapsulation, and inheritance.	Supports OOP features like polymorphism, encapsulation, and inheritance.
Relationship	C++ is often considered a superset of C, but this is not strictly true due to some incompatible features.	Can use most of C's features but has additional features that are not compatible with C.
Keywords	Has a smaller set of keywords (around 32).	Has more keywords (63 in standard C++), including OOP-related ones.
Data Types	Supports basic built-in data types.	Supports both built-in and user-defined data types.
File Extension	Usually uses .c for source files.	Typically uses .cpp for source files.
Access Modifiers	Does not have access modifiers like public, private, protected	Has access modifiers: public, private, protected
Standard I/O Header	Uses <stdio.h> for standard I/O operations.	Uses <iostream> (without .h) for standard I/O operations.
Information Hiding	No inherent support for information hiding.	Supports information hiding through encapsulation.
Focus	More process or method-focused.	More data-focused, but also supports process-focused programming.
Exception Handling	Lacks direct support for exception handling.	Supports exception handling.
Standard I/O Operations	Uses scanf() and printf() for standard I/O.	Uses cin and cout for standard I/O, alongside C-style I/O.
Function & Operator Overloading	Does not support function or operator overloading.	Supports both function and operator overloading.
Main Function	The main() function cannot be called from other functions.	Similar to C, main() cannot be called from other functions.
Reference Variables	Does not support reference variables.	Supports reference variables.



Parameter	C	C++
Programming Paradigm	C is a structural or procedural programming language.	C is a structural as well as an object-oriented programming language.
History	C was developed by scientist Dennis Ritchie in 1972 at Bell Laboratories.	C was developed by Bjarne Stroustup in 1979.
Approach	C follows a top-down approach	C follows the bottom-up approach.
Keywords	C contains 32 keywords	C++ contains 63 keywords.
Data Types	C supports built-in data types.	C++ support both built-in and user-defined data types.
File Extension	.c is the file extension for C programming language	.cpp is the file extension for C++ programming language
Header File	<stdio.h> header file is used by C language	<iostream.h> header file is used by C++ language
Allocation and Deallocation of Memory	In C language, we use calloc() and malloc() for dynamic allocation of memory and free() for deallocation of memory.	In C++ language, we use a new operator for the allocation of memory and a delete operator for the deallocation of memory.



Access Modifier	C language does not support access modifier	C++ support access modifier
Security	C does not have any security features so it can be manipulated by outsider	C++ is a secure language as it offers security features such as data hiding and encapsulation
Reference Variable	C does not support reference variable	C++ support reference variable
Function Overloading and Function Overriding	C don't supports function overloading and function overriding	C++ supports function overloading and function overriding
Exception Handling	C does not support exception handling directly, it uses the function that support exception handling	C++ directly support exception handling with the help of try – catch block
Program Division	C is a procedural language, so code written in C are divided in separate blocks known as function	C++ is a object oriented language, so code and divided into classes and objects
Inline Function	C doesn't support inline function	C++ support inline function
Driven Type	C is known as function driven language	C is known as object driven language



Compatibility	Code written in C language can be run on C++ compiler as C is the foundational language	Code written in C++ language can be run on C compiler as C++ language includes OOP's concept
Data and Function	In C, the data and function are separated as it is a procedural programming language	In C++, the data and function are encapsulated as it is a object oriented programming language
Input and Output Function	In C scanf() and printf() functions are used to take the input and output respectively	In C++ cin and cout functions are used to take the input and output respectively
Application Development	C language is more suitable for low level implementation such as network driver, text editor, assembler, etc	C++ language is more suitable for high level implementation such as game development, smartwatches, embedded systems, etc
Namespace	To prevent the collision and organize the code, namespace is needed but C does not support that	C++ support the namespace
Used By	MySQL, Windows Kerne, Oracle Database, Telegram messenger, etc	Google Chrome, Torque 3-D game, Microsoft Office, etc

Summary

In this lesson, you should have learned how to:

- What is C C++ ?
- Why do we need C++ ?
- Basic Concepts of C++
- Differences Between C & C++