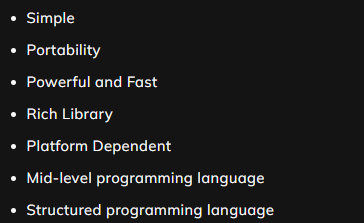
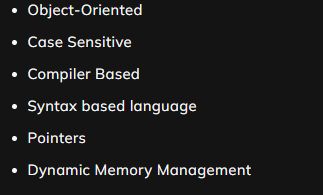
## Features of C++



C++ provides a lot of **features** that are given below.

1) Simple: C++ is a simple language because it provides a structured approach (to break the problem into parts), a rich set of library functions, data types, etc. It allows us to follow both procedural as well as functional approach to design our flow of control.

2) Portability: It is the concept of carrying the instruction from one system to another system. In C++, Language .cpp file contains source code, and we can also edit this code. .exe file contains the application, which is the only file that can be executed. When we write and compile any C++ program on the Windows operating system, it efficiently runs on other window-based systems.

3) Powerful: C++ is a very powerful programming language, and it has a wide variety of data types, functions, control statements, decision-making statements, etc. C++ is a fast language as compilation and execution time is less. Also, it has a wide variety of data types, functions & operators.

4) Rich Library: C++ library is full of in-built functions that save a tremendous amount of time in the software development process. As it contains almost all kinds of functionality, a programmer can need in the development process. Hence, saving time and increasing development speed.

5) Platform Dependent: Platform dependent language means the language in which programs can be executed only on that operating system where it is developed & compiled. It cannot run or execute on any other operating system. E.g., compiled programs on Linux won’t run on Windows.

6) Mid-level programming language: C++ can do both low-level & high-level programming. That is the reason why C++ is known as a mid-level programming language.

7) Structured programming language: C++ is a structured programming language as it allows to break the program into parts using functions. So, it is easy to understand and modify.

8) Object-oriented: C++ is an object-oriented programming language. OOPs make development and maintenance easier, whereas, in Procedure-oriented programming language, it is not easy to manage if code grows as project size grows. It follows the concept of oops like polymorphism, inheritance, encapsulation, abstraction.

9) Case sensitive: C++ is a case-sensitive programming language. In C++ programming, 'break and BREAK' both are different.

10) Compiler Based: C++ is a compiler-based language, unlike Python. C++ programs used to be compiled, and their executable file is used to run it due to which C++ is a relatively faster language than Java and Python.

11) Syntax-based language: C++ is a strongly typed syntax-based programming language. If any language follows the rules and regulations strictly, it is known as a strongly syntax-based language. Other examples of syntax-based languages are C, C++, Java, .net etc.

12) Pointer: C++ supports pointers that allow the user to deal directly with the memory and control the programmer. This makes it very suitable for low-level tasks and very complicated projects. It is known to increase the speed of execution by decreasing the memory access overhead.

13) Dynamic Memory Management: It supports the feature of dynamic memory allocation. In C++ language, we can free the allocated memory by calling the free() function. These features are missing in languages like C.

## Uses of C++

There are several benefits of using C++ because of its features and security; below are some uses of C++ Programming Language:

**Operating Systems:**One of the key requirements of an Operating System is that it should be very fast as it is responsible for scheduling and running the user programs. The strongly typed and fast nature of C++ makes it an ideal candidate for writing operating systems. Also, C++ has a vast collection of system-level functions that also help in writing low-level programs**.**Microsoft Windows or Mac OS X, or Linux - all operating systems have some parts programmed in C++.

**Games:**Again, since most of the games need to be faster to support smooth game play, C++ is extensively used in game design. C++ can easily manipulate hardware resources, and it can also provide procedural programming for CPU-intensive functions.

**Browsers:**With the fast performance of C++, most browsers have their rendering software written in C++. Browsers are mostly used in C++ for rendering purposes. Rendering engines need to be faster in execution as most people do not like to wait for the web page to be loaded.

**Libraries:** Many high-level libraries use C++ as the core programming language. For example, TensorFlow uses C++ as the back-end programming language. Such libraries required high-performance computations because they involve multiplications of huge matrices to train Machine Learning models. As a result, performance becomes critical. C++ comes to the rescue in such libraries.

**Graphics:**C++ is widely used in almost all graphics applications that require fast rendering, image processing, real-time physics, and mobile sensors.

**Cloud/Distributed Systems:** Cloud storage systems use scalable file-systems that work close to the hardware; also, the multi-threading libraries in C++ provide high concurrency and load tolerance.

**Embedded Systems:**C++ is closer to the hardware level, and so it is quite useful in embedded systems as the software and hardware in these are closely coupled. Many embedded systems use C++, such as smartwatches, MP3 players, GPS systems, etc.

**Compilers:**Compilers of various programming languages use C++ as the back-end programming language.

## main() function

Look at the following piece of code:

**int main() {**

**Statement 1;**

**Statement 2;**

**...**

**}**

Starting with the line:

### int main():

**int:** This is the return type of the function. You will get this thing clear once you reach the Functions topic.

**main():** This is the portion of any C++ code inside which all the commands are written and executed.

This is the line at which the program will begin executing. This statement is similar to the start block of flowcharts.

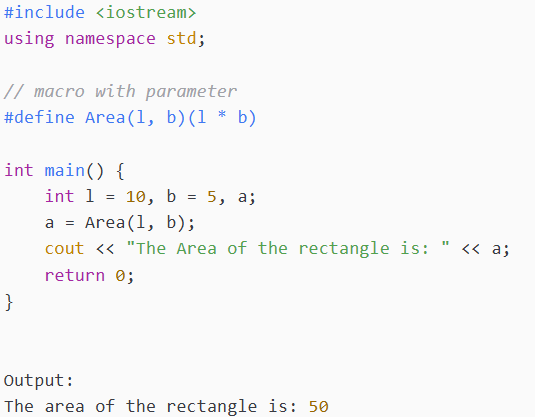
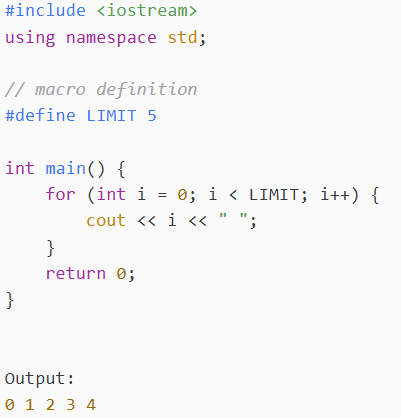
As you will move further in the course, you will get a clear glimpse of this function. Till then, just note that you will have to write all the programs inside this block.

**{}:** all the code written inside the curly braces is said to be in one block, also known as a particular function scope. Again, these things will be clear when you will study functions.

## Macros

Macros are a piece of code in a program that is given some name. Whenever the compiler encounters this name, the compiler replaces the name with the actual piece of code. The ‘#define’ directive is used to define a macro.

Note: There is no semicolon(‘;’) at the end of the macro definition.



### Comments

C++ comments are hints that a programmer can add to make their code easier to read and understand. They are completely ignored by C++ compilers.

There are two ways to add comments to code:

**//        -** Single Line Comment

**/\* \*/    -**Multi-line Comments

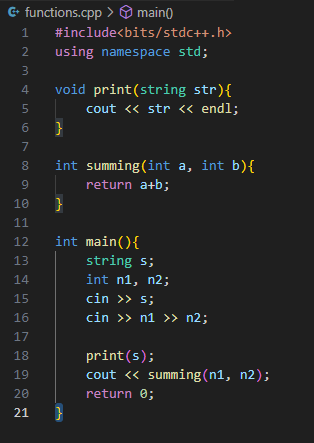
**Example: Single line comment. Example: Multi-line Comment.**

## Patterns

1. For the outer loop, count the number of lines
2. For the inner loop, focus on the columns and connect them somehow to the rows.
3. Print the “@” inside the inner For Loop
4. Observe symmetry (optional)

## Functions:



## STL: Standard Template Library

* Algorithms
* Containers
* Functions
* Iterators

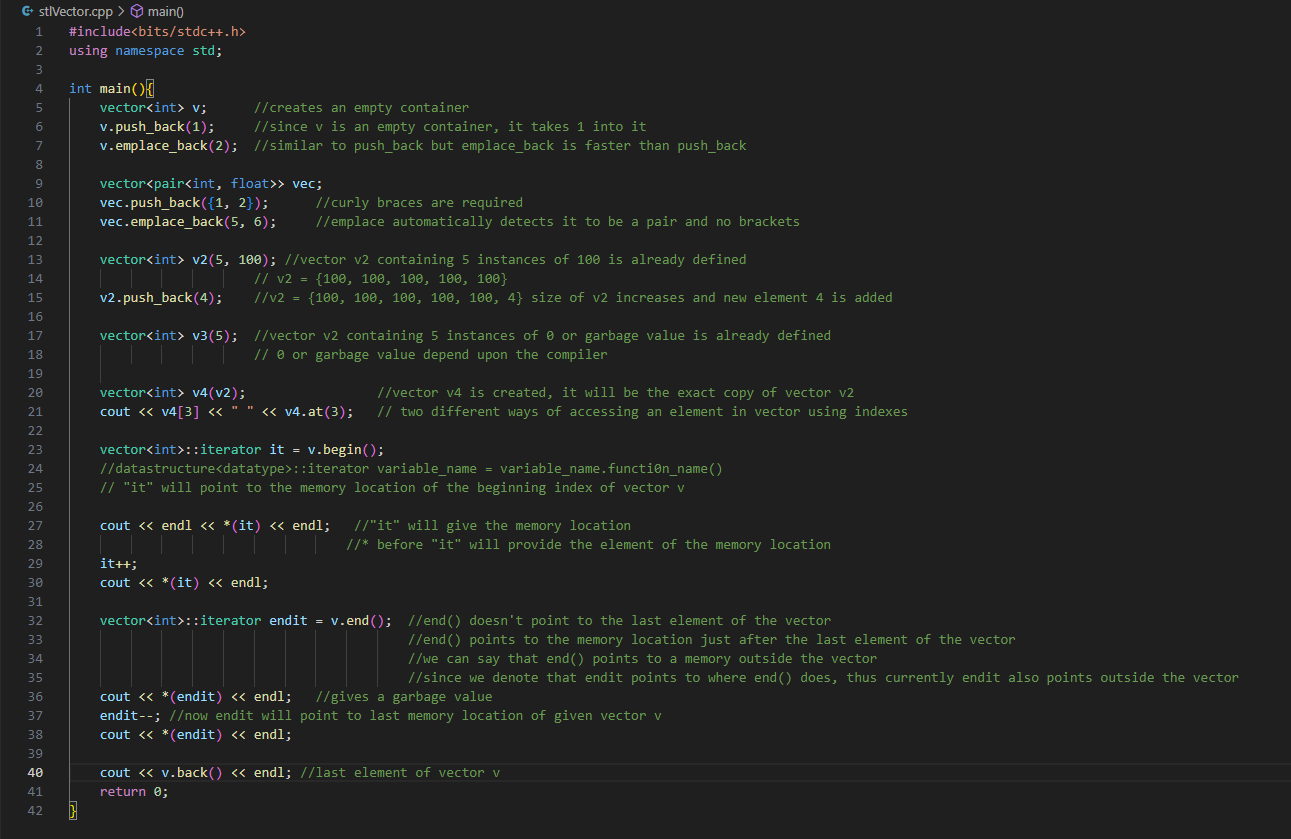
### Pair:

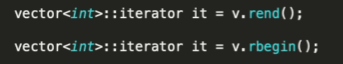
### 

Pair lies inside the utility library.

### Vectors:

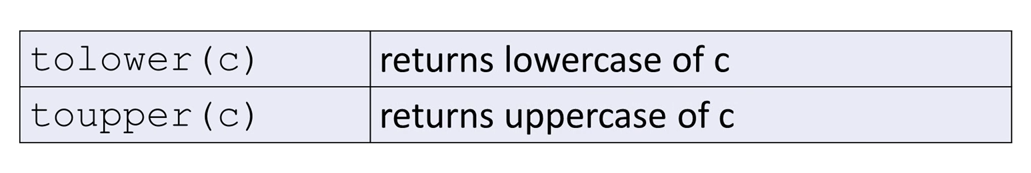
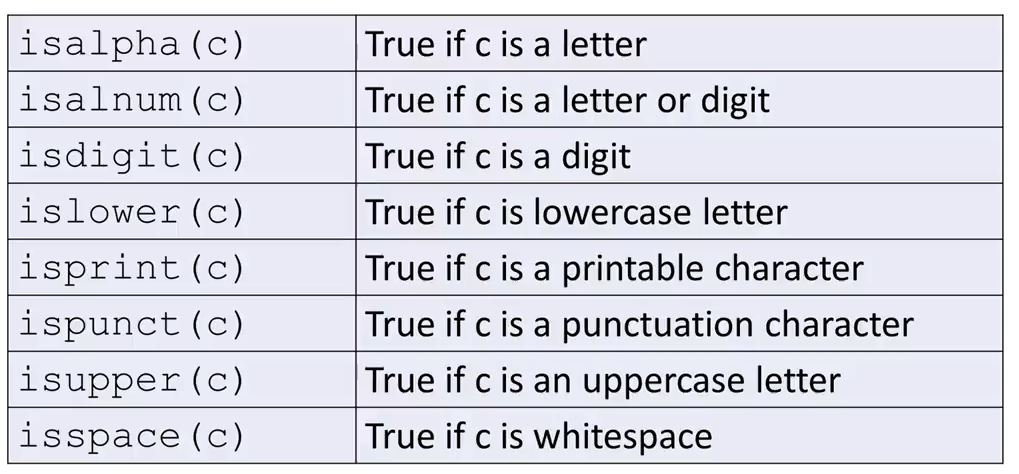
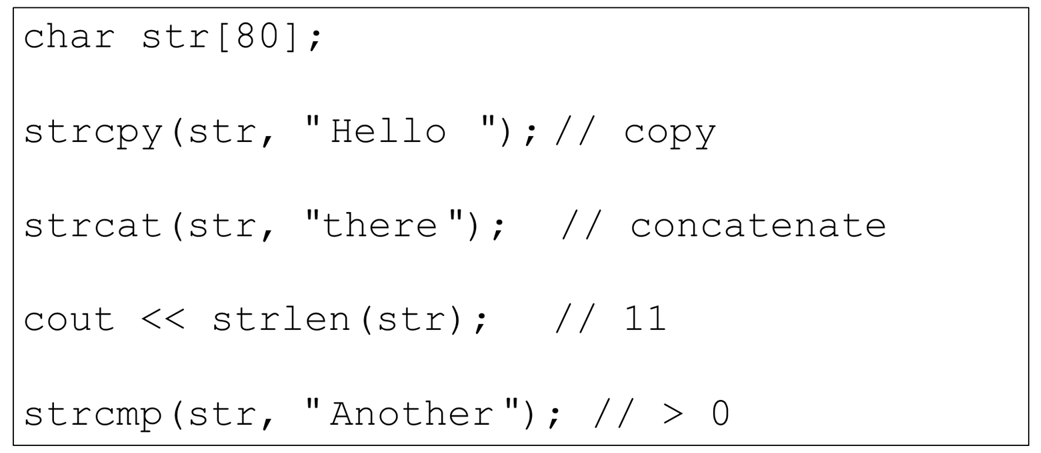
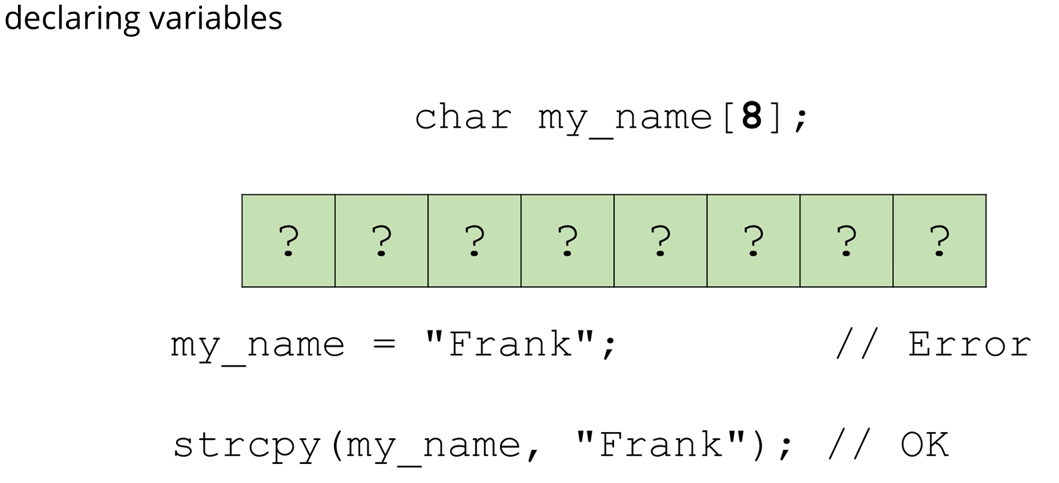
Vector is a container that is dynamic in nature, we can always increase the size of this container when needed later in the program.



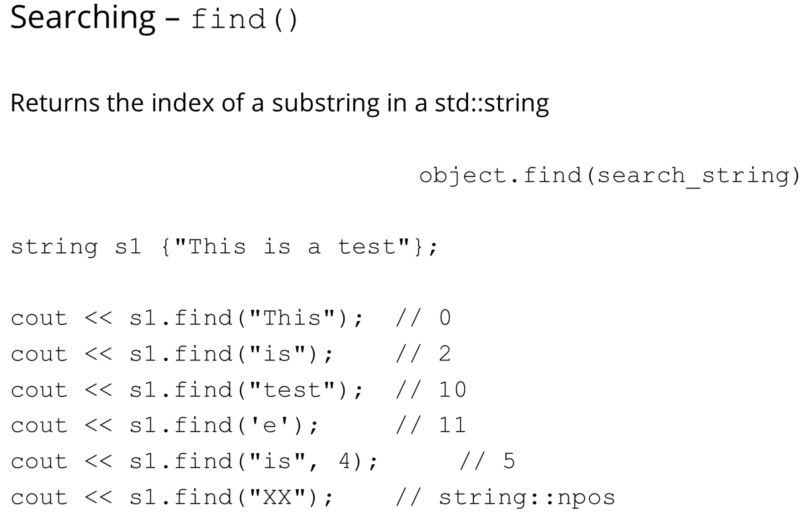
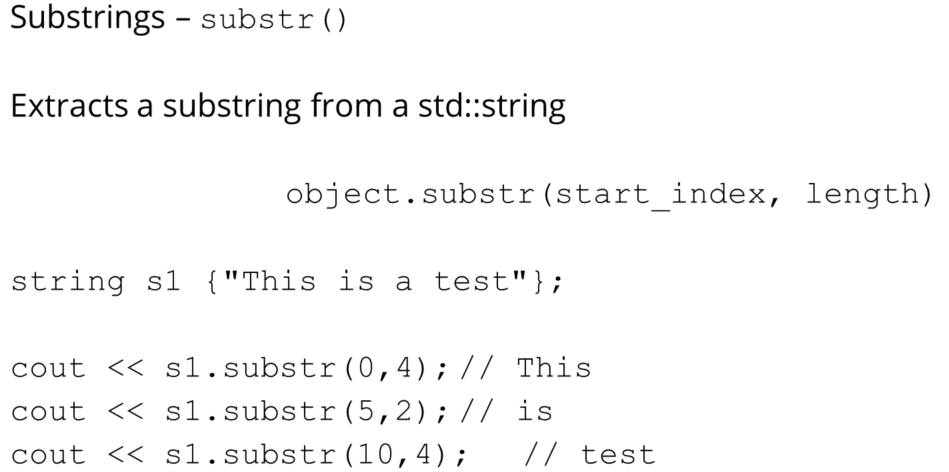
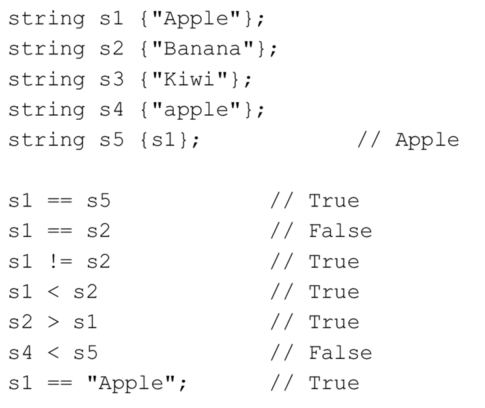
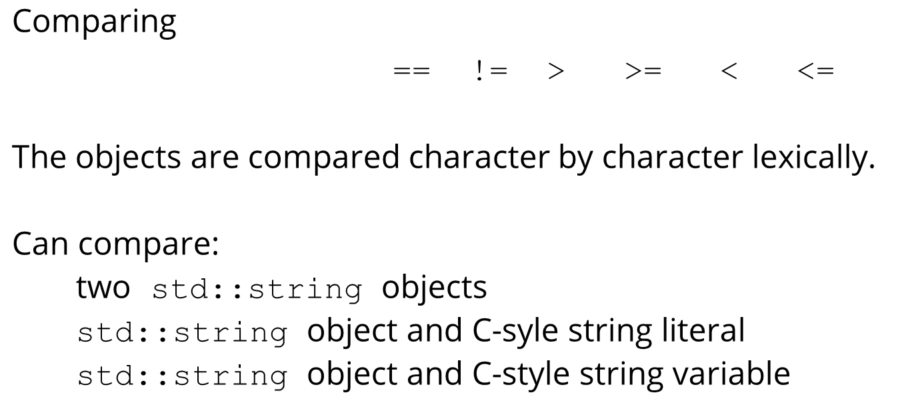
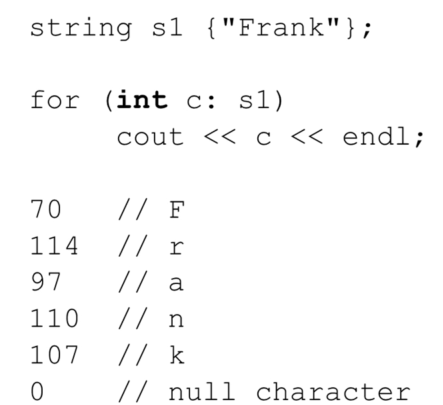
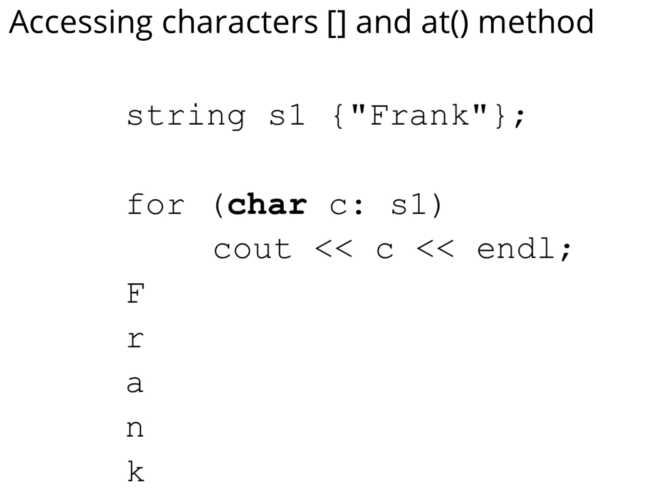
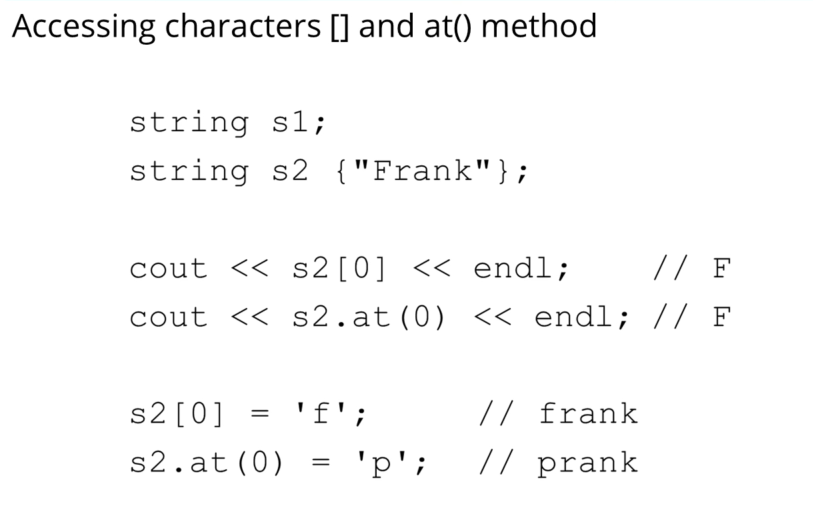
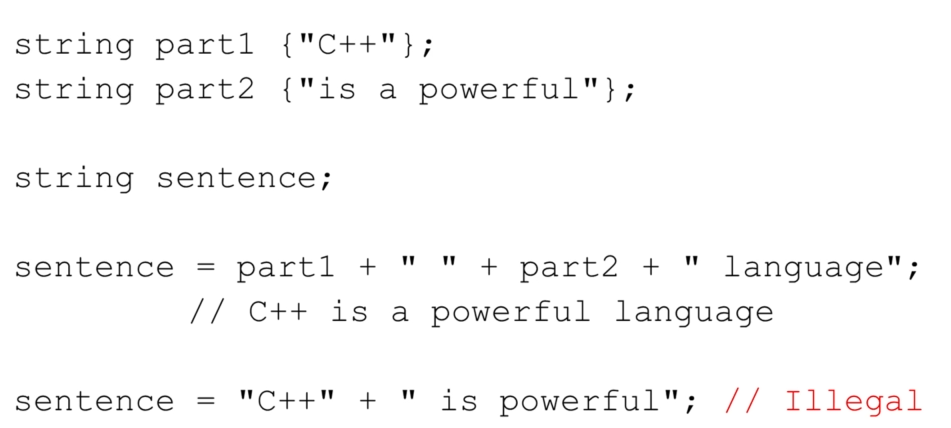
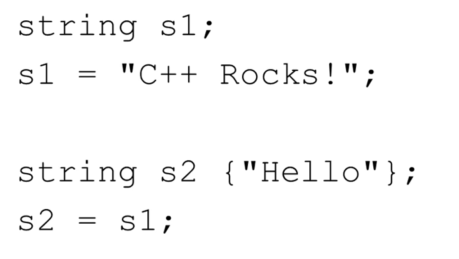
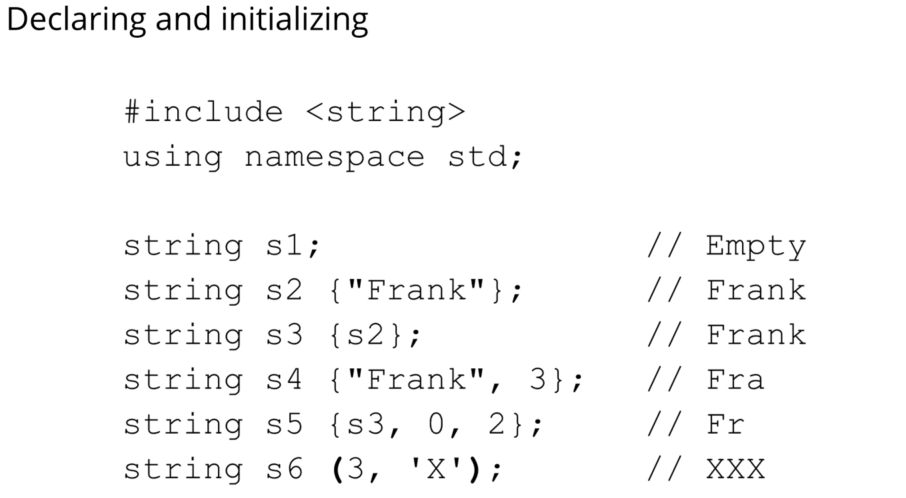
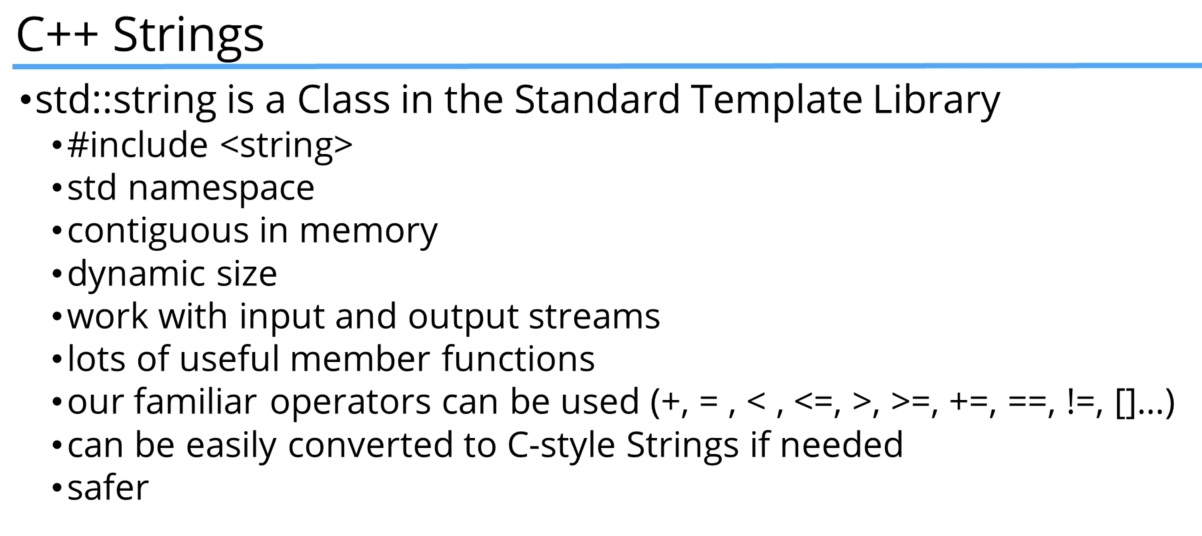


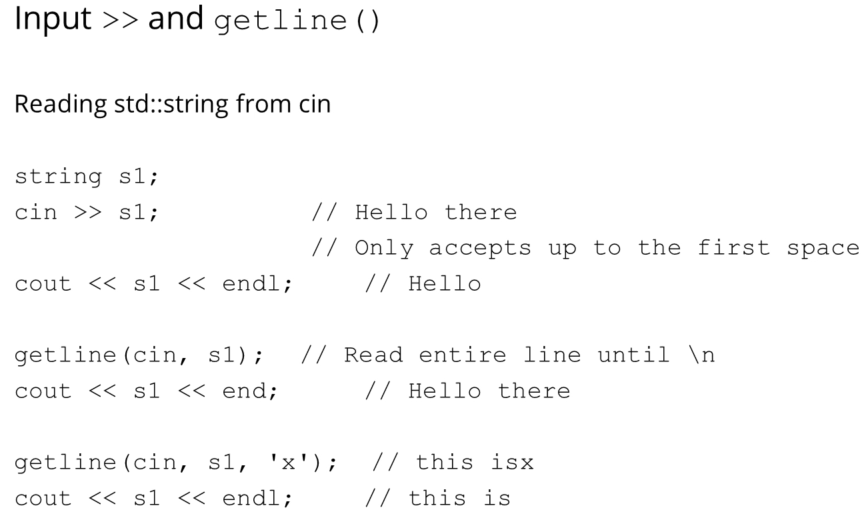
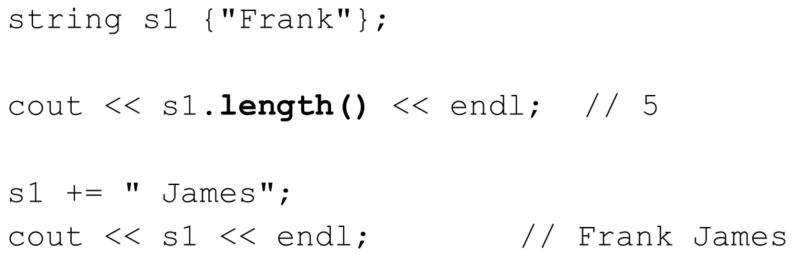
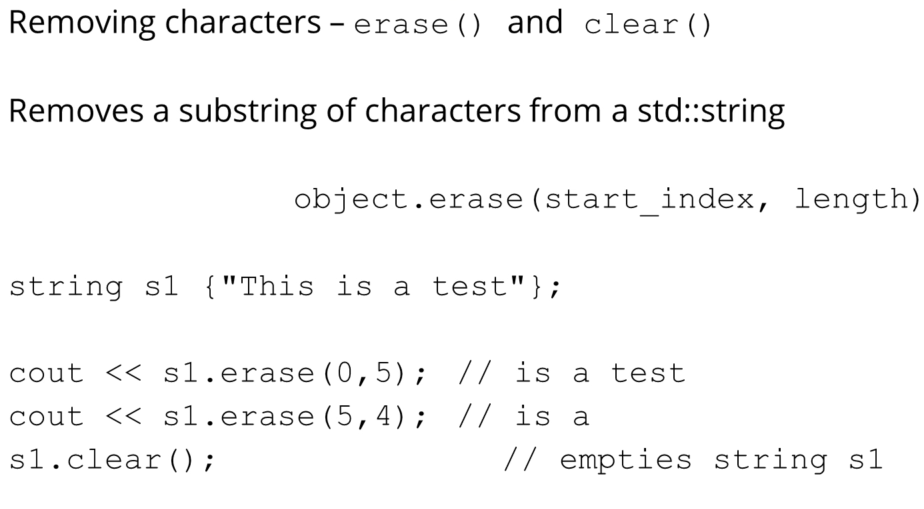
## C-Style Strings

### 



## C++ Strings





### Basic Maths

Whenever there is division…

* if the division is happening by 10, then time complexity is O(log10(n)); n being the number divided by 10
* if the division is happening by 2, then time complexity is O(log2(n)); n being the number divided by 2
* and so on similarly….

If the number of iterations is depending upon division of a number, then time complexity will be logarithmic… the way explained above

#### Euclidean Algorithm for GCD/HCF

Time complexity

### Recursion

When a function calls itself until a specific condition is met

Segmentation Fault

Stack Overflow

Infinite Recursion

Waiting function calls

Base condition

Recursion Tree

Backtracking

### Hashing

Hash Array

Number Hashing

Character Hashing

ASCII Values

Map/Hash Map

Taking “n” as number of elements  
Time Complexity of Map:  
 storing and fetching in all best, average and worst take O(log(n)) of time complexity

Time Complexity of Unordered Map:  
 storing and fetching in best and average case take O(1) and in worst(this case is very rare) case take O(n) of time complexity

Prefer unordered map because of less time complexity, only prefer ordered map if time limit exceeds

Worst case happens because of internal collision

Hashing  
 division method  
 folding method  
 mid square method

Collision

### Sorting:

Arranging data in required order

#### Selection Sort

1. Select: minimum and swap it to the first play
2. In the left-over array, again select the minimum and swap it to the second place, so on till the end

#### Bubble Sort

Pushes the maximum to the last by adjacent swapping, compare adjacent values and swap them if second is bigger than first

#### Insertion Sort

It always takes and element and places it in its correct position

#### Merge Sort

Divide and Merge

#### Quick Sort

1. Pick a pivot and place it in its correct place in its sorted array
2. Smaller on the left and larger on the right

### Array

When we initialise an array inside MAIN, then an array with garbage values is created

When we initialise an array outside MAIN, i.e. in GLOBAL, then an array with all values equal to Zero is created

Contagious memory location

### Binary Search