**What is inline function :**  
The inline functions are a C++ enhancement feature to increase the execution time of a program. Functions can be instructed to compiler to make them inline so that compiler can replace those function definition wherever those are being called. Compiler replaces the definition of inline functions at compile time instead of referring function definition at runtime.   
NOTE- This is just a suggestion to compiler to make the function inline, if function is big (in term of executable instruction etc) then, compiler can ignore the “inline” request and treat the function as normal function.

#include<iostream>

using namespace std;

inline int cube(int x){

return x\*x\*x;

}

int main(){

cout<<cube(2);

return 0;

}

**How is it different from memory allocated to normal variables?**  
For normal variables like “int a”, “char str[10]”, etc, memory is automatically allocated and deallocated. For dynamically allocated memory like “int \*p = new int[10]”, it is programmers responsibility to deallocate memory when no longer needed. If programmer doesn’t deallocate memory, it causes [memory leak](http://www.geeksforgeeks.org/what-is-memory-leak-how-can-we-avoid/) (memory is not deallocated until program terminates).

**What are memory management operators?**

There are two types of memory management operators in C++:

* new
* delete

*New operator:*

* The new operator in C++ is used for dynamic storage allocation. This operator can be used to create object of any type.

*General syntax of new operator in C++:*

pointer variable = new datatype;

* In the above statement, new is a keyword and the pointer variable is a variable of type datatype.

*or example:*

1. int \*a = new int;

2. \*a = 20;  
      **or**  
3.  int \*a = new int(20);

* In the above example, the **new** operator allocates sufficient memory to hold the object of datatype int and returns a pointer to its starting point.
* the pointer variable a holds the address of memory space allocated.

*delete operator:*

* The **delete** operator in C++ is used for **releasing memory** space when the object is no longer needed.
* Once a new operator is used, it is efficient to use the corresponding delete operator for release of memory.

*General syntax of delete operator in C++:*

delete pointer variable;

**#include <iostream>**

**void main() {**

**//Allocates using new operator memory space in memory for storing a integer datatype**

**int \*a= new int;**

**\*a=100; cout << " The Output is:a= " << \*a;**

**//Memory Released using delete operator**

**delete a;**

**}**

# Formatted and Unformatted Data

Data which is received by the program without any modifications and sent to the output device without any modifications is known as unformatted data. On the other hand, sometimes we may want to apply some modifications to the actual data that is being received or sent. For example, we might want to display an integer in hexadecimal format in the output, leave some whitespace when printing a number and adjustments in the decimal point. Such modified data in known as formatted data.

As an example for formatted data, if we want to display a decimal number in hexadecimal format, we can use the **hex manipulator** as shown below:

cout<<hex<<15

inner class:

#include<iostream>

using namespace std;

class outer{

public:

class inner{

public:

void m1(){

cout<<"hello";

}

};

};

int main(){

outer::inner in;

in.m1();

return 0;

}