Introduction to C++ and differences between C and C++ language

C++ Hierarchy

- ALGOL(1960)
- BCPL(1967)
- B(1970)
- Traditional C(1972)
- K&R C(1978)
- ANSI C(1989, 99….)
- C With Classes
- C++(1984)
- C++ 98, 14, 17,20 …

Features of C++

- C++ is a general-purpose programming language that was developed as an enhancement of the C Language to include an Object Oriented-Paradigm. It is an imperative and compiled language. C++ has a number of features, including:
  - Object-Oriented Programming
  - Simple and popular
  - High-Level Language
  - Strongly typed
  - Case-sensitive
  - Memory Management (Dynamic Memory Allocation)
  - Multi-threading support

Introduction

- C++ is derived from C Language. It is a Superset of C.
- Earlier C++ was known as C with classes.
- In C++, the major change was the addition of classes and a mechanism for inheriting class objects into other classes.
- Most C Programs can be compiled in C++ compiler.
- C++ expressions are the same as C expressions.
- All C operators are valid in C++.

Structure of Cpp program

Comments(documentation)

Preprocessor Statements/ Header File Inclusion

Function Declaration

Global Variable Declaration

Extern Variable Declaration

Structure / Class Declaration

Function definitions

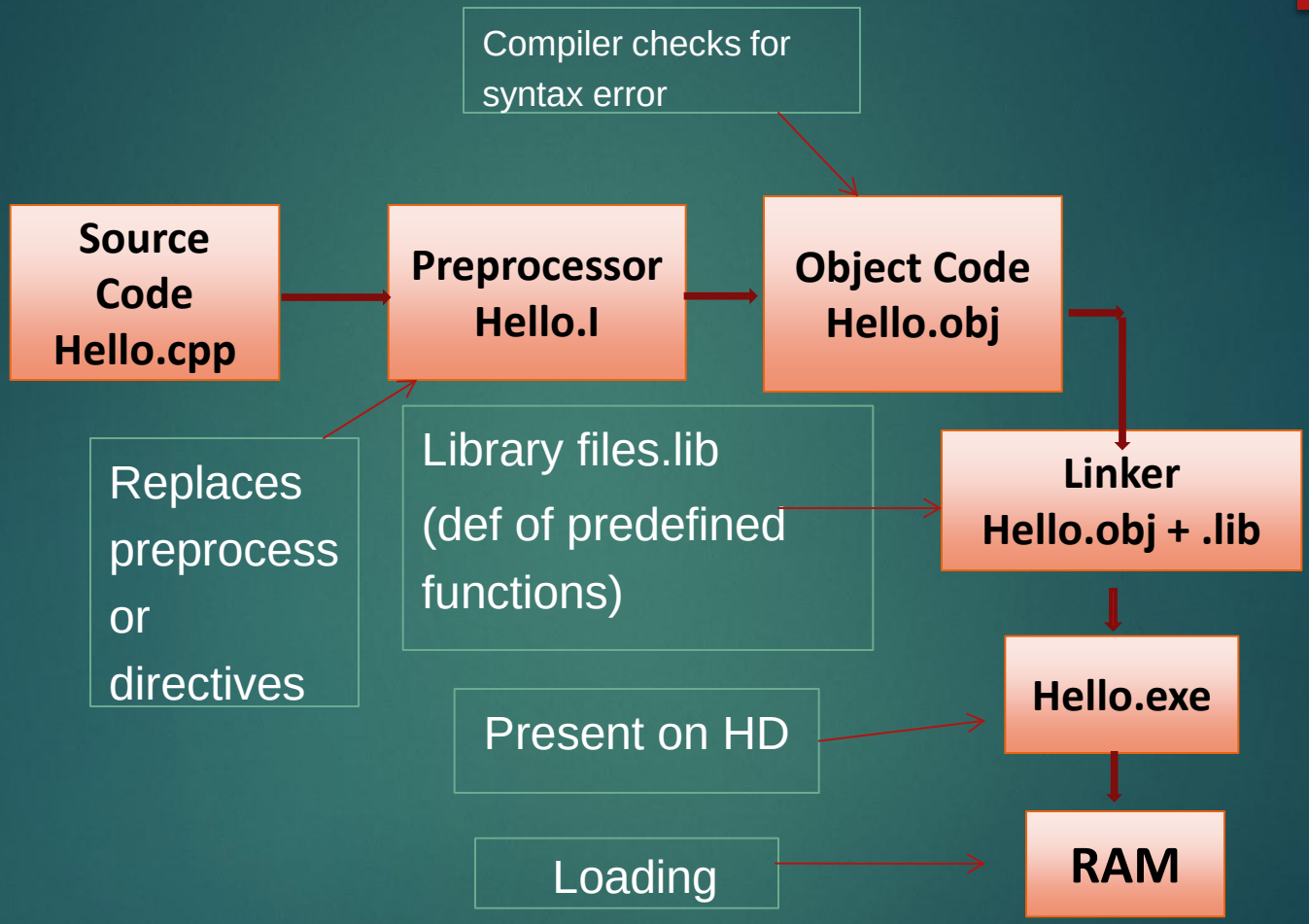
Hello World in C++


```
#include<iostream>
using namespace std;
int main()
{
cout<<"Hello World";
return 0;
}
```

Compilation:
on Linux Platform
g++ Hello.cpp
After compilation it will produce a.out file
Execution: ./a.out

- cout is object of ostream
- cin is object of istream
- **>>** called as **extraction** or input operator
- **<<** is called as **insertion** operator or output operator
- Using namespace **std** is used to specify namespace
- **returning 0 tell OS that program execution is graceful**

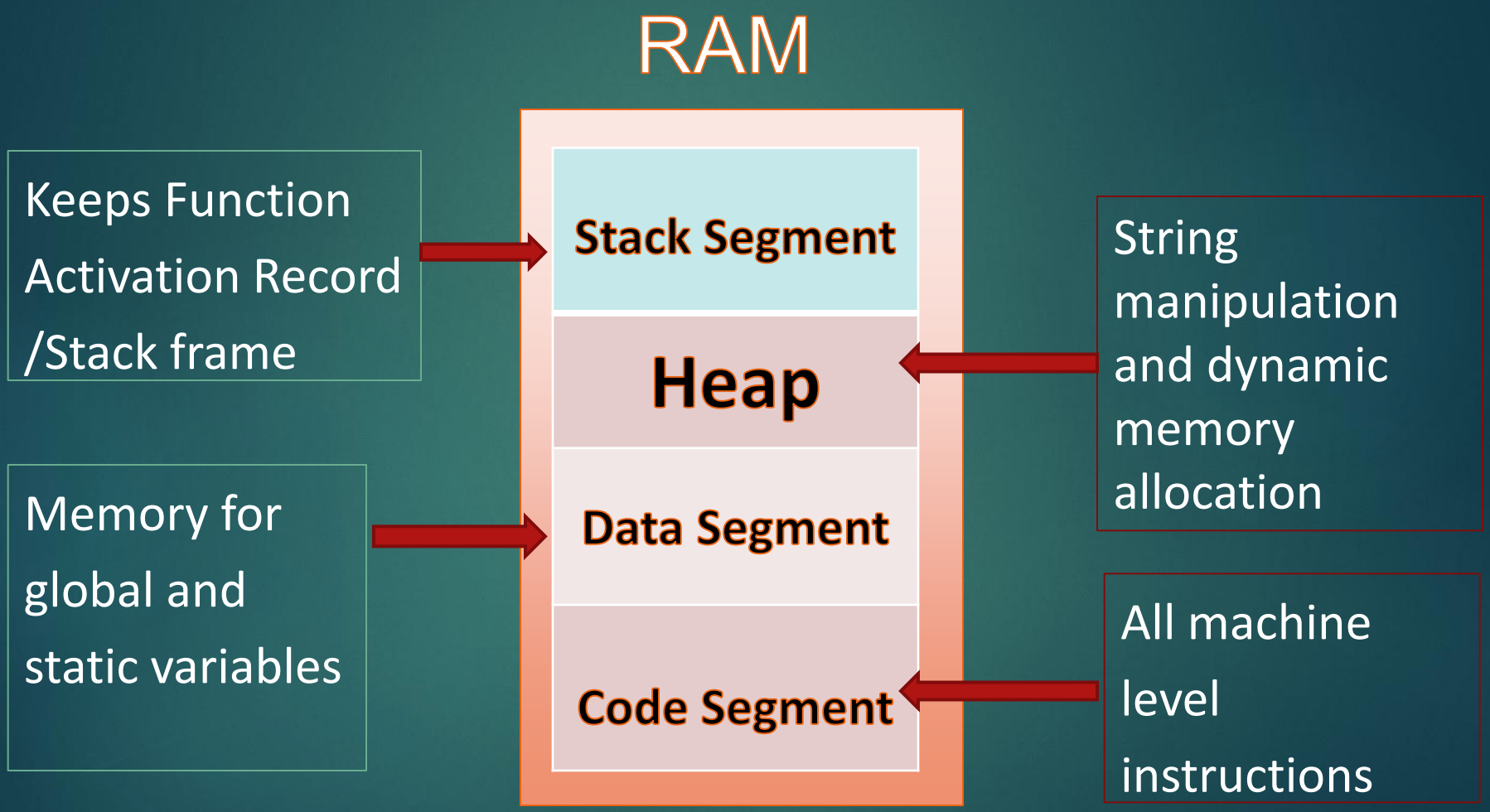
Compilation process



Program memory

- When we run a program ,os allocates part of memory for that program and then copies executable from disk to memory.
- The C++ compiler divides this area in 4 parts
  - Stack
  - Heap
  - Code Area/Segment
  - Data Area/Segment
    - Initialized data
    - Uninitialized data

Program Memory



Code Segment

- Fixed in nature because size of program is known at load time.
- **It is reserved for executable code of program**.
- This is read only memory area we can not change it during execution.
- Only Pointers to functions can access this area.

Data Segment

- Fixed in nature because size of programs data is known at load time.
- It is contains internal and external static variables, global variables, initialized array and structures and constant strings.
- **Initialized Data Area:**
  - **On Initialization static and global variables are stored at initialized data area.** All others variables gets stored in uninitialized data area.
- **Uninitialized Data Area:**
  - In uninitialized data area variables get initialized to 0 nut in initialized area they are initialized with their respective values.
  - **Static and global variables known as load time variables.**

Keywords from C to C++

auto const double float int short
struct unsigned break continue else for
long signed switch void case default
enum goto register sizeof typedef
volatile char do extern if return
static union while

Keywords in C++

30 reserved words that were not in C:

asm public private protected new
delete bool false true try catch
throw class this friend template
using namespace inline typename
typeid wchar_t explicit virtual
mutable operator static_cast
dynamic_cast reinterpret_cast const_cast

Predefined identifiers

cin endl INT_MIN iomanip main
npos std **cout** include INT_MAX
iostream MAX_RAND NULL string

Difference between C and C++

C Programming Language	C++ Programming Language
C is Procedural Language	C++ is Object Oriented Language. It supports Object Oriented principle like Encapsulation, Inheritance, Polymorphism, Coupling and Cohesion. C++ is treated as not pure Object Oriented Lang. Reason: We can write C++ program without writing class and creating OBJECT and Friend functions Example: ***#include<iostream>*** ***using namespace std;*** ***int main()*** ***{*** ***cout<<"\nHello World\n";*** ***return 0;*** ***}***

Difference between C and C++

C Programming Language	C++ Programming Language
Top down approach is used in Program Design. i.e. Program to functions	Bottom up approach adopted in Program Design i.e. Objects to functionality
Variable declaration/definition is allowed only at the start of block(Scope) Example: ***#include<stdio.h>*** ***int main()*** ***{ int a,b,c;*** ***a = 10;*** ***b = 10;*** ***c = a+b;*** ***printf("Sum=%d",c);*** ***return 0;*** ***}***	Variable declaration/definition is allowed anywhere in block(scope) Example: ***#include<iostream>*** ***using namespace std;*** ***int main()*** ***{int a,b;*** ***a=10;*** ***b=10;*** ***int c = a+b;*** ***cout<<c;*** ***return 0;*** ***}***

Difference between C and C++

C Programming Language	C++ Programming Language
In C, malloc() and calloc() Functions are used for Memory Allocation and free() function for memory Deallocating.	In C++, new and delete operators are used for Memory Allocating and Deallocating
In C, malloc,calloc, realloac are functions and we need to include stdlib.h header to use them.	In C++, new and delete are part of language as these are operators in C++, we dont need any header to be include to use them.
In C, malloc,calloc and realloc does not understand datatypes. These functions returns void* Example: *int *p= **(int*)malloc(sizeof(int));*** Note: we are passing no of bytes to be allocated to malloc	In C++, new operator understand datatype, it allocates memory and calls contrsuctor for datatype and return specific pointer Example: *int * p = **new int;*** Note: We are just specifying what is datatype.

Difference between C and C++

C Programming Language	C++ Programming Language
In C, no special memory deallocation of array Example: For single int *int *p = (int*)malloc(sizeof(int));* *free(p);* For array of 10 int *int *p = (int*)malloc(10*sizeof(int));* *free(p);*	In C, special memory deallocation of array Example: For single int *int *p = new int;* *delete p;* For array of 10 int *int *p = new int[10];* *delete []p;*
In C, we can call main() Function through other Functions	In C++, we can call main() Function through other functions but compiler gives warning. warning: implicit declaration of function `int main(...)'

Difference between C and C++

C Programming Language	C++ Programming Language
In C, there is no mechanism for scope resolution ***int i=10;*** ***int main()*** ***{int i=20;*** ***i = i+i;*** ***printf("\n%d",i);*** ***return 0;*** ***} // print 40***	In C++, there is scope resolution operator which help in specifying scope ***#include<iostream>*** ***using namespace std;*** ***int i=10;*** ***int main()*** ***{int i=20;*** ***i = i+ ::i;*** ***cout<<i;*** ***return 0;*** ***} // print 30***
In C, User defined constant can not be used as Array size Example: ***const int s=10;*** ***int arr[s]; // compilation error***	In C, User defined constant can be used as Array size as C++ constants are pure constants Example: ***const int s=10;*** ***int arr[s]; // allowed***

Difference between C and C++

C Programming Language	C++ Programming Language

C constants are compile time constants they can be changed at run time.
Example 1: (Compile time modification check)
const int k=10;
k= k+1; //Compilation error
Example 2: (Run time modification check)
#include<stdio.h>
int main(){
const int k=10;
int* p =(int*)&k;
****p=100;***
printf("\n%d",*p); // will print 100
printf("\n%d",k); // will print 100
return 0;}
	C++ constants are true constants they cannot be changed at run time.
	Example 1: (Complie time modification check)
	const int k=10;
	k= k+1; //Compilation error
	Example 2: (Run time modification check)
	#include<iostream>
	using namespace std;
	int main(){
	const int k=10;
	int* p =(int*)&k;
	****p=100;***
	cout<<*p<<endl; // will print 100
	cout<<k<<endl; // will print 10
	return 0;}

Difference between C and C++

C Programming Language	C++ Programming Language
C is weakly typed language **int i = 10;** **char c =i**; // compiler will generate WARNING for type conversion(casting)	C++ is strongly typed language **int i = 10;** **char c =i;** // compiler will generate ERROR for type conversion(casting)
C uses pointers for memory handling	C++ minimizes use of pointers by introducing references

In C, function call cannot be written on LHS of assignment operator.	In C++, function call can be written on LHS of assignment operator.
	#include<iostream>
	using namespace std;
	int k = 10;
	int& function()
	{ return k;}
	int main(){
	function()=100;
	cout<<k; // Will print 100
	return 0;}

Difference between C and C++

C Programming Language	C++ Programming Language

Placeholder areguments in functions are not allowed.	Function can have placeholder arguments
	Example:
	int function(int a, int b, int)
	{
	return a+b;
	}

Namespaces are not allowed to separate scope.	Namespaces can be used to separate out scope.
String handling is done using char array	String handling can be done using char array but C++ provides simple to use string datatype to handle strings.
C struct does not support Encapsulation	C++ struct supports Encapsulation

The C++ string class

- Must #include <string> to create and use string objects
- Can define string variables in programs

```
string name;
```

- Can assign values to string variables with the assignment operator

```
name = “Kareena";
```

- Can display them with cout

```
cout << name;
```

- Can input string with cin

```
cin >>name;
```

Thank You

Feel the difference!!!! ……C++