

BASICS OF C++



TWO VERSIONS OF C++

- A traditional-style C++ program -

```
#include <iostream.h>

int main()
{
    /* program code */
    return 0;
}
```

TWO VERSIONS OF C++ (CONT.)

- A modern-style C++ program that uses the new-style headers and a namespace -

```
#include <iostream>
using namespace std;

int main()
{
    /* program code */
    return 0;
}
```

SIMPLE C++ PROGRAM

```
include <iostream>

int main()
{
    char str[16];
    cout << "Enter a string: ";
    cin >> str;
    cout << "You entered: " << str;
    return 0;
}
```

THE NEW C++ HEADERS

- The new-style headers do not specify filenames.
- They simply specify standard identifiers that might be mapped to files by the compiler, but they need not be.
 - `<iostream>`
 - `<vector>`
 - `<string>`, not related with `<string.h>`
 - `<cmath>`, C++ version of `<math.h>`
 - `<cstring>`, C++ version of `<string.h>`
- Programmer defined header files should end in “.h”.

APPLICATIONS OF C++

- Adobe Systems. Most of the major **applications** of adobe systems are developed in **C++**
- Google **Applications**. ...
- Mozilla Firefox and Thunderbird.
- MySQL Server.
- Bloomberg RDBMS.
- Games:
- Graphic User Interface (GUI) based applications:
- Web Browsers
- Database Software
- Operating Systems
- Enterprise Software
- Compilers:

STRUCTURE & CLASS IN C++

- In C++, a structure is same as class except the following differences:
- Members of a class are private by default and members of struct are public by default.
- When deriving a struct from a class/struct, default access-specifier for a base class/struct is public.
- And when deriving a class, default access specifier is private.

COMPILING & LINKING

- When the compiler begins to run it first runs a preprocessor phase.
- During that phase the text from the header files are inserted in to source code.
- Translates it to machine code.
- File containing Machine code called an object file and has the file extension .obj.
- The process of translating the source code into an object file is called compiling.
- Now, the object file created is in the binary form.

- Hence, the compilation phase generates the relocatable object program and this program can be used in different places without have to compile again.
- But you still can't run these object files until to convert them into executable file, now here linker comes into play, which links all the object files to generate single executable file.

LINKING

- After the compiler has created all the object files, another program is called to bundle them into an executable program file.
- That program is called a linker and the process of bundling them into the executable is called linking.
- The linker looks at all the object files you have told it to use.
- Linking as the name suggests, refers to creation of a single executable file from multiple object files.
- The file created after linking is ready to be loaded into memory and executed by the system .

- There is difference in linking and compilation when it comes to understanding errors.
- Compiler shows errors in syntax, for example semi-colon not mentioned, data type not defined etc. but if there is an error that function has been defined multiple times, then this error is from linker as its indicating that two or more source code files have the same meaning and that is leading to an error.