Basic C++ Program:

```
#include<iostream>

using namespace std;

int main(){
   cout<<"Namaste Dunia"<<endl;
   return 0;
}
```

Output: PS E:\C++ PROGRAMMES> g++ main.cpp
PS E:\C++ PROGRAMMES> ./a.exe
Namaste Dunia

Let us split the code:

#include<iostream>

It is a preprocessor directive and includes the input/output stream library (iostream). It allows you to use functionality related to input and output in your C++ program.

using namespace std;

The std namespace contains various standard C++ libraries, including the cout object used for output.

int main(){

The main function, which is the entry point of every C++ program. The execution of the program starts from here.

cout << "Namaste Dunia" << endl;

The cout object from the std namespace to print the string "Namaste Dunia" to the console.

The << operator is used to send the string to the standard output (console).

endl is a manipulator that represents a newline character, ensuring that the cursor moves to the next line after printing the string.

return 0;

This line indicates that the program execution was successful, and it returns a status code of 0 to the operating system. A return value of 0 conventionally signifies that the program executed without errors.

Home Work Questions:

1. What is preprocessor(#include)
2. What is mean by std::cout instead of using “using namespace std”
3. What will happen we write “return -1” instead of “return 0”
4. What is another way to print a new line in c++ instead of “endl”

1.What is preprocessor(#include)

The preprocessor in C++ is used for processing the code before it is compiled by the compiler. It does many tasks such as including files, conditional compilation, using macros, etc. The preprocessor also allows the developers to select which portions of code should be included or excluded.

The code processed by the preprocessor is called expanded code and is generally saved with a ".i" file extension.

Preprocessor Directives in C++

In C++, the preprocessor directives are special commands that are used to instruct the preprocessor. It begins with a '#' symbol and tells the preprocessor to the modify source code before compilation. There are different preprocessor directives in C++ for different operations.

To Read more: https://www.geeksforgeeks.org/cpp-preprocessors-and-directives/

2.What is mean by std::cout instead of using "using namespace std"

The **cout** is a predefined object of ostream class, and it is used to print the data on the standard output device. Generally, when we write a program in Linux operating system for G++ compiler, it needs "std" namespace in the program.We use it by writing **using namespace std;** then we can access any of the objects like cout, cin.

```
// Program to show the use of cout
// without using namespace

#include <iostream>
int main()
{
        std::cout << "GeeksforGeeks";
        return 0;
}
```

std:cout: A namespace is a declarative region inside which something is defined. So, in that case, cout is defined in the std namespace. Thus, std::cout states that is cout defined in the std namespace otherwise to use the definition of cout which is defined in std namespace. So, that std::cout is used to the definition of cout from std namespace.

```
// Program to show use of using namespace

#include <iostream>

using namespace std;

int main()

{
```

```
    cout << "GeeksforGeeks";
    return 0;
}
```

To Read more: https://www.geeksforgeeks.org/difference-between-cout-and-stdcout-in-c/

3.What will happen we write “return -1” instead of “return 0”

In C++, the main function is expected to return an integer value to the operating system. A return value of 0 conventionally indicates that the program has executed successfully, while a non-zero value usually signifies an error or an abnormal termination.

If we write return 0; to return -1;, it means that the program is indicating an abnormal termination or an error condition to the operating system. The specific interpretation of the return value may vary depending on the operating system or the environment in which the program is executed.

For many operating systems, a non-zero return value from the main function suggests that the program did not complete successfully. This can be useful for scripting or other programs that invoke your C++ program to check whether it executed correctly.

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

int main(){
   cout << "Namaste Dunia" << endl;
   return -1;  // Changed from return 0;
}
```

If we run this program and inspect the return value (e.g., by checking the value of the special variable $? in Unix-like environments after running the program), it will likely be a non-zero value, indicating that the program did not terminate normally.

In practice, it is a good practice to use return 0; to indicate successful execution unless you have a specific reason to return a different value to convey information about an error or specific condition.

4.What is another way to print a new line in c++ instead of “endl”

The another way is using “\n”

Program:

```
#include<iostream>
using namespace std;
int main(){
    cout<<"Namaste Dunia\n";
    cout<<"Welcome";
    return 0;
}
```

```
Output: PS E:\C++ PROGRAMMES> g++ helloworld.cpp
        PS E:\C++ PROGRAMMES> ./a.exe
        Namaste Dunia
        Welcome
        PS E:\C++ PROGRAMMES>
```