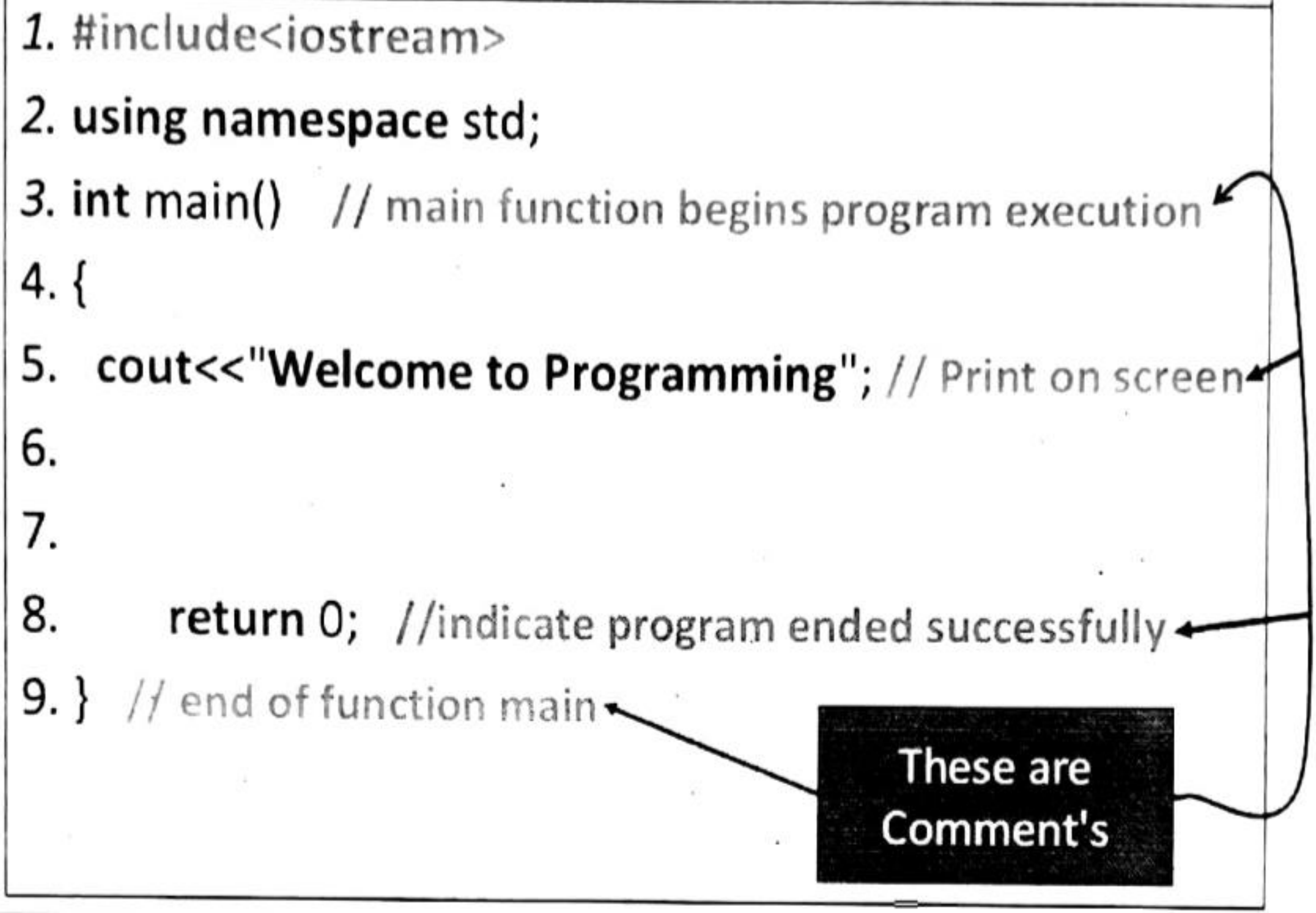
C++ Program Structure

First C++ Program

```
#include<iostream>
using namespace std;
int main()   // main function begins program execution
{
 cout<<"Welcome to Programming"; // Print on screen


      return 0;   //indicate program ended successfully
}  // end of function main
```

First C++ Program



```
#include<iostream>
using namespace std;
int main()    // main function begins program execution
{
 cout<<"Welcome to Programming"; // Print on screen


     return 0;   //indicate program ended successfully
}  // end of function main
```

These are Comment's

- Two slash signs indicate that the rest of the line is a comment inserted by the programmer but which has no effect on the behavior of the program. Programmers use them to include short explanations or observations concerning the code or program. In this case, it is a brief introductory description of the program.

First C++ Program

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Welcome to Programming";

    return 0;
}
```

Preprocessor Directive

tells the compiler to include a service for 'Stream I/O'

#include <iostream>

- beginning with a hash sign (#) are directives read and interpreted by what is known as the *preprocessor*. They are special lines interpreted before the compilation of the program itself begins. In this case, the directive #include <iostream>, instructs the preprocessor to include a section of standard C++ code, known as *header iostream*, that allows to perform standard input and output operations, such as writing the output of this program (Welcome to Programming) to the screen.

First C++ Program

```
#include<iostream>
using namespace std;
int main()
{
        cout<<"Welcome to Programming";

        return 0;
}
```

Use Standard namespace

tells the compiler to use the "standard namespace", Namespaces are a mechanism for avoiding naming conflicts in large Programs

First C++ Program

```
#include<iostream>
using namespace std;
int main()
{
        cout<<"Welcome to Programming";

        return 0;
}
```

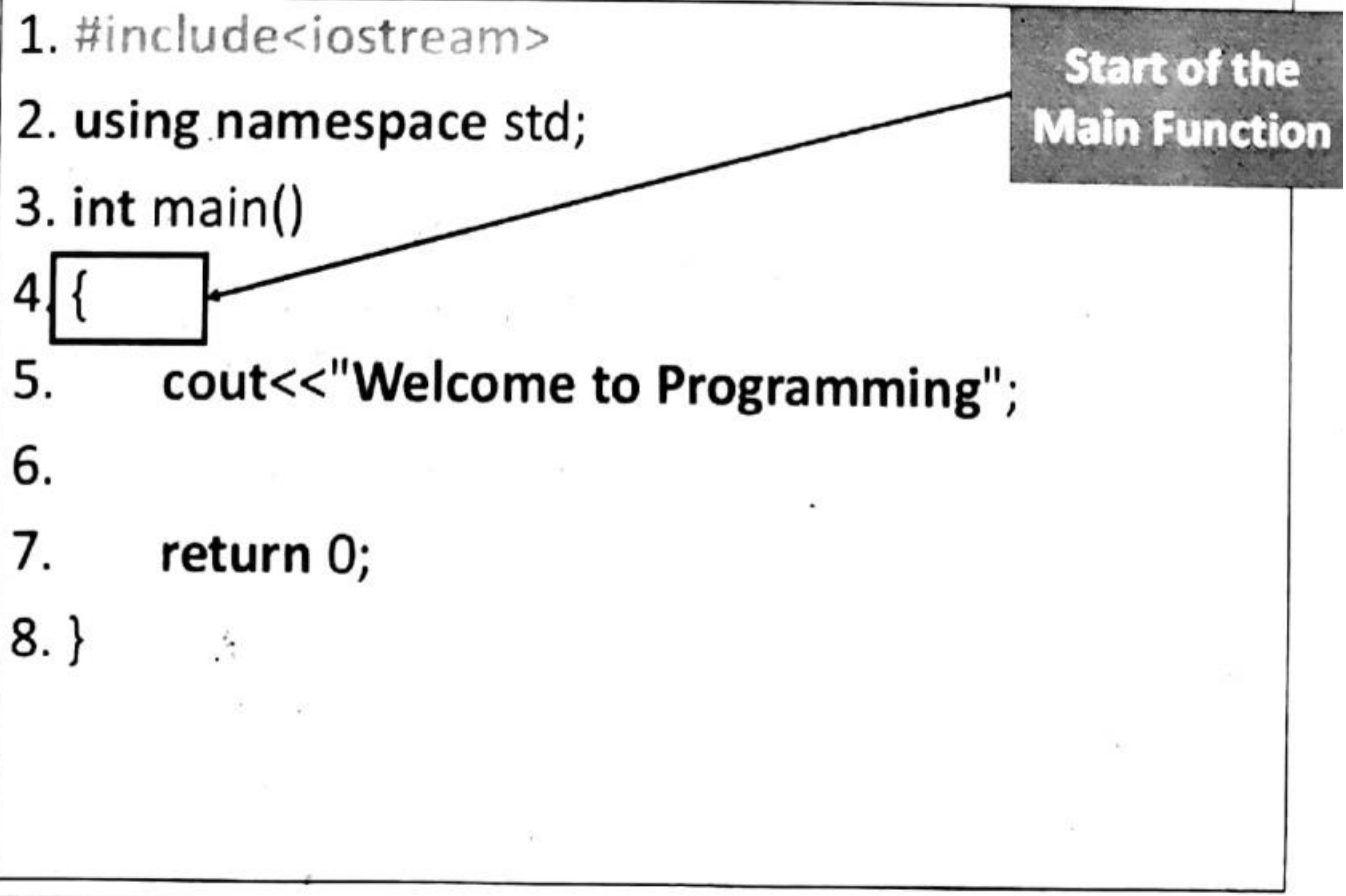
Main() Function

int main ()

- int main ()This line initiates the declaration of a function. Essentially, a function is a group of code statements which are given a name: in this case, this gives the name "main" to the group of code statements that follow. Functions will be discussed in detail in a later chapter, but essentially, their definition is introduced with a succession of a type (int), a name (main) and a pair of parentheses (()), optionally including parameters.

  The function named main is a special function in all C++ programs; it is the function called when the program is run. The execution of all C++ programs begins with the main function, regardless of where the function is actually located within the code.

First C++ Program



```
#include<iostream>
using namespace std;
int main()
{
        cout<<"Welcome to Programming";

        return 0;
}
```

Start of the Main Function

- { and }The open brace ({) indicates the beginning of main's function definition, and the closing brace (}), indicates its end. Everything between these braces is the function's body that defines what happens when main is called. All functions use braces to indicate the beginning and end of their definitions.

Blocks in C++

- A *block* (or a ***compound statement***) is a group of statements surrounded by **braces** { }.
- **All the statements** inside the block is **treated as one unit**. Blocks are used as the *body* in constructs like function, if-else and loop, which may contain multiple statements but are treated as one unit.
- **For Example**

```
int main()
{
    ...statements...
}
```

This is one Block or Unit