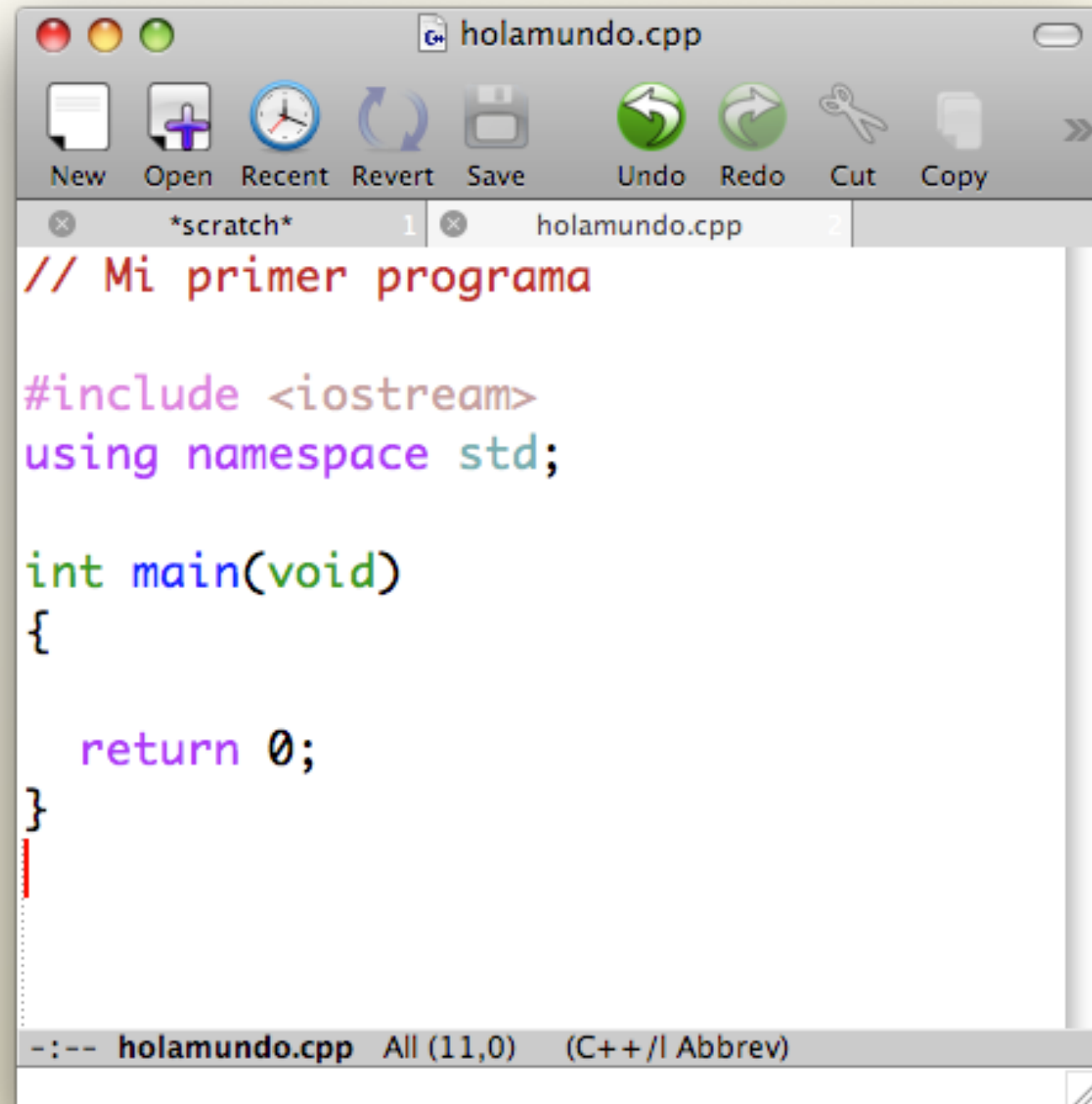


# Program Structure and Compilation

William Oquendo, [woquendo@gmail.com](mailto:woquendo@gmail.com)

Credits: Computational Physics - Landau and Paez, Wikipedia,  
[www.cplusplus.com/doc/tutorial](http://www.cplusplus.com/doc/tutorial) , Schaum series-Programming  
in C++

# Plantilla



The image shows a screenshot of a C++ IDE window titled "holamundo.cpp". The window has a standard macOS-style title bar with red, yellow, and green buttons. Below the title bar is a toolbar with icons for New, Open, Recent, Revert, Save, Undo, Redo, Cut, and Copy. The main editing area contains the following C++ code:

```
// Mi primer programa

#include <iostream>
using namespace std;

int main(void)
{

    return 0;
}
```

The code is color-coded: comments are red, preprocessor directives are purple, and keywords are green. The main function is enclosed in curly braces. A vertical red line is visible at the end of the closing brace of the main function.

At the bottom of the window, there is a status bar that reads: "-:-- holamundo.cpp All (11,0) (C++/I Abbrev)".

# Template

Comentario de una linea

Directiva de  
preprocesador

Usar el espacio de nombres  
standar

Funcion principal

Retorno de la funcion (final)

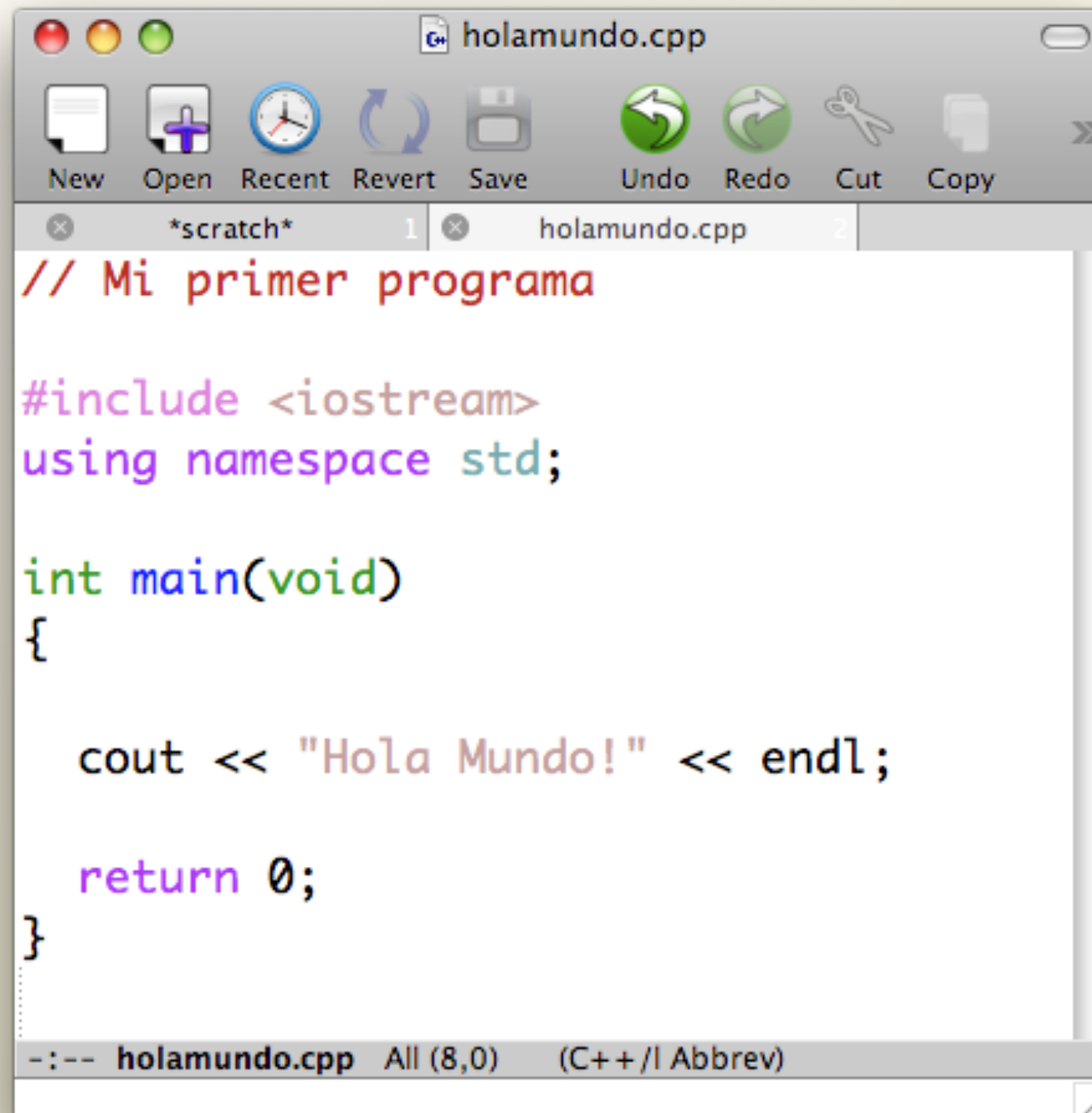
```
// Mi primer programa

#include <iostream>
using namespace std;

int main(void)
{
    return 0;
}
```

The screenshot shows a code editor window with a menu bar (New, Open, Recent, Revert, Save, Undo, Redo, Cut, Copy) and a tab bar with two tabs: '\*scratch\*' and 'holamundo.cpp'. The code is color-coded: comments are red, preprocessor directives are purple, namespace names are blue, and the main function signature is green. Arrows from external labels point to the following parts of the code: 'Comentario de una linea' points to the first line; 'Directiva de preprocesador' points to the #include line; 'Usar el espacio de nombres standar' points to the using namespace std; line; 'Funcion principal' points to the int main(void) line; and 'Retorno de la funcion (final)' points to the return 0; line. The status bar at the bottom shows '-:-- holamundo.cpp All (11,0) (C++/I Abbrev)'.

# Hola mundo



The image shows a screenshot of a C++ IDE window titled 'holamundo.cpp'. The window has a standard macOS-style title bar with red, yellow, and green buttons. Below the title bar is a toolbar with icons for New, Open, Recent, Revert, Save, Undo, Redo, Cut, and Copy. The main editing area contains the following C++ code:

```
// Mi primer programa

#include <iostream>
using namespace std;

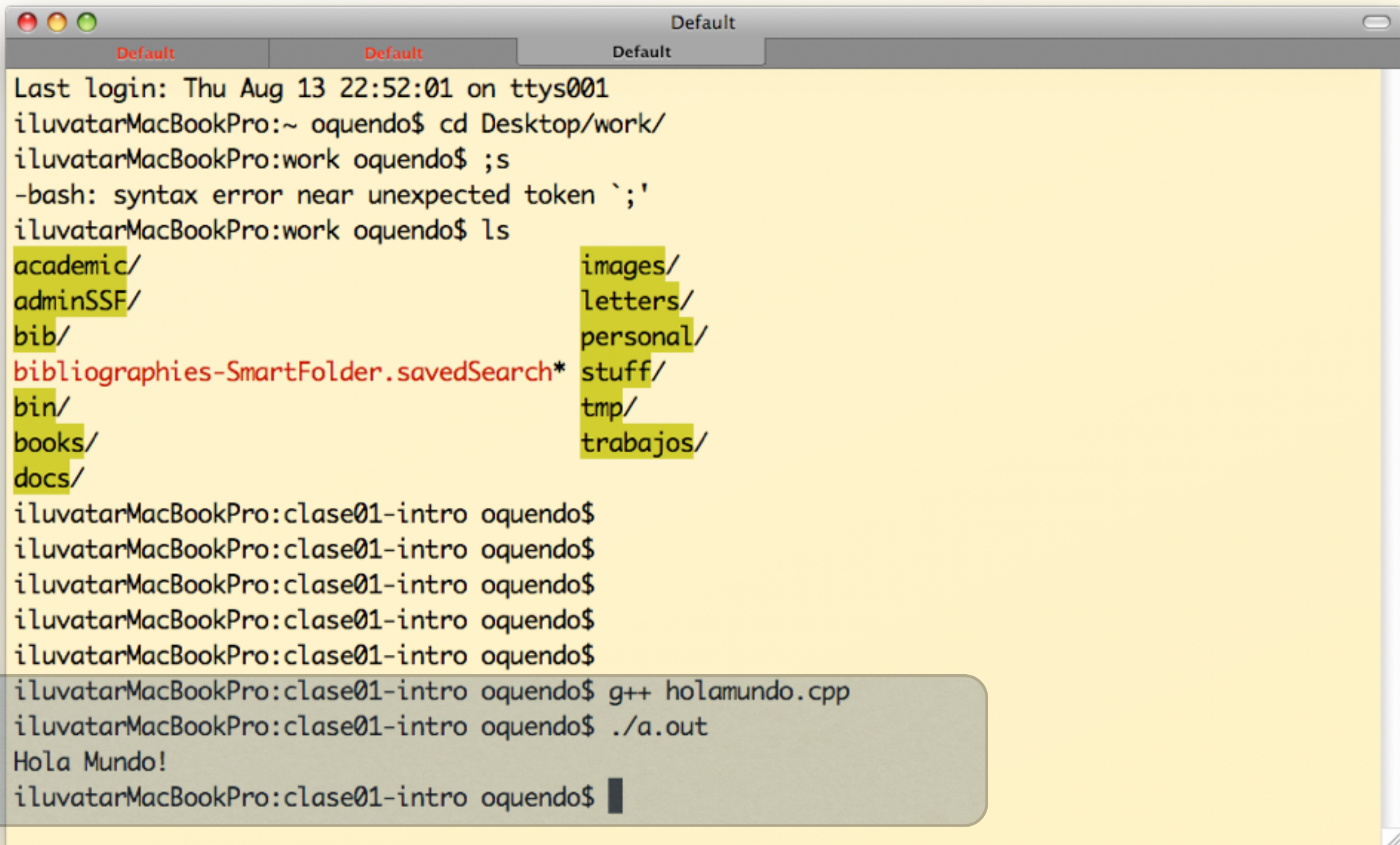
int main(void)
{

    cout << "Hola Mundo!" << endl;

    return 0;
}
```

At the bottom of the window, a status bar displays the file name 'holamundo.cpp', the cursor position 'All (8,0)', and the language mode '(C++ /I Abbrev)'.

# Compiling and running

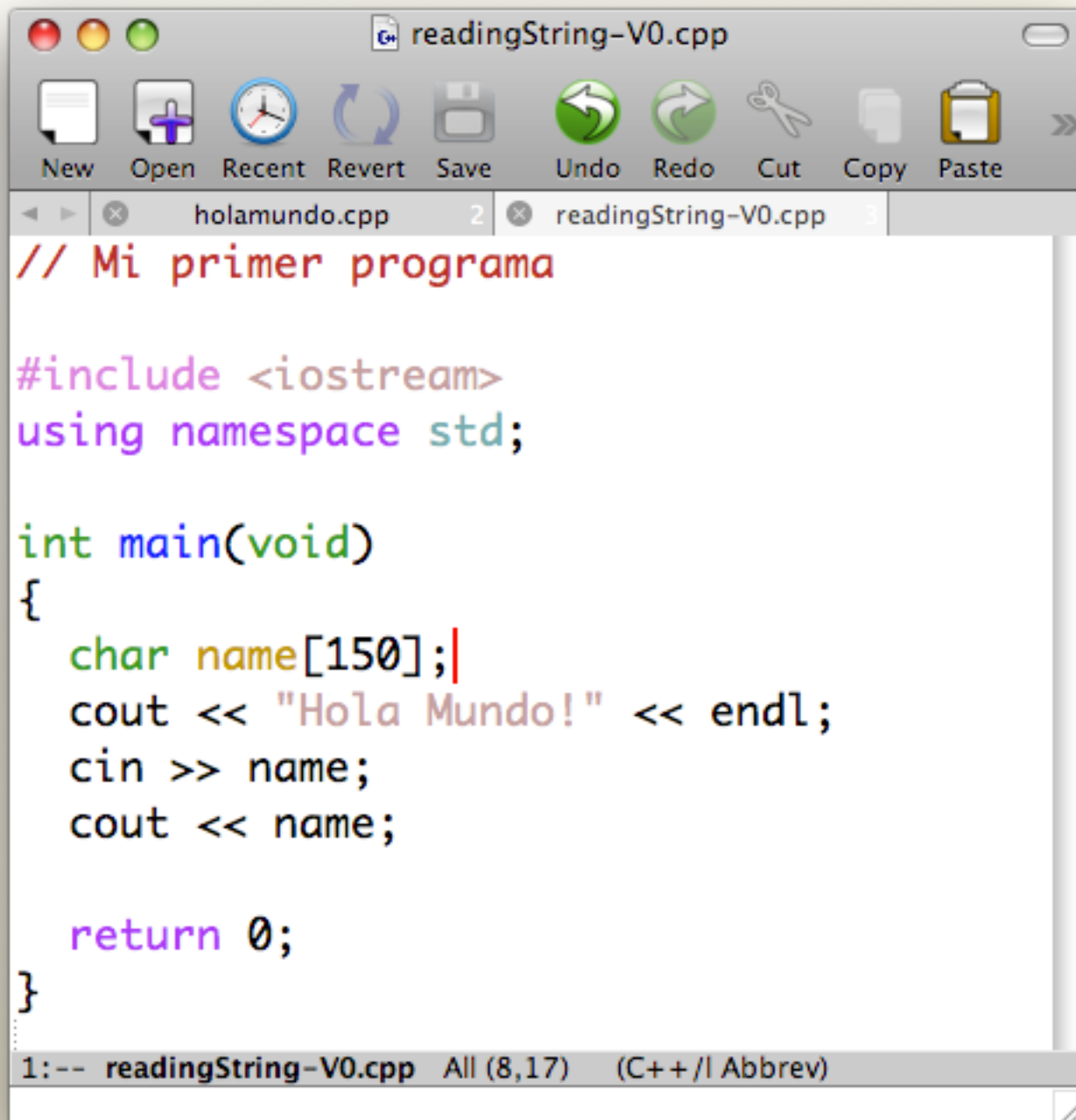


```
Default
Default
Default
Last login: Thu Aug 13 22:52:01 on ttys001
iluvatarMacBookPro:~ oquendo$ cd Desktop/work/
iluvatarMacBookPro:work oquendo$ ;s
-bash: syntax error near unexpected token `;'
iluvatarMacBookPro:work oquendo$ ls
academic/          images/
adminSSF/          letters/
bib/               personal/
bibliographies-SmartFolder.savedSearch* stuff/
bin/               tmp/
books/             trabajos/
docs/

iluvatarMacBookPro:clase01-intro oquendo$
iluvatarMacBookPro:clase01-intro oquendo$
iluvatarMacBookPro:clase01-intro oquendo$
iluvatarMacBookPro:clase01-intro oquendo$
iluvatarMacBookPro:clase01-intro oquendo$
iluvatarMacBookPro:clase01-intro oquendo$ g++ holamundo.cpp
iluvatarMacBookPro:clase01-intro oquendo$ ./a.out
Hola Mundo!
iluvatarMacBookPro:clase01-intro oquendo$
```



# Exercise



```
// Mi primer programa

#include <iostream>
using namespace std;

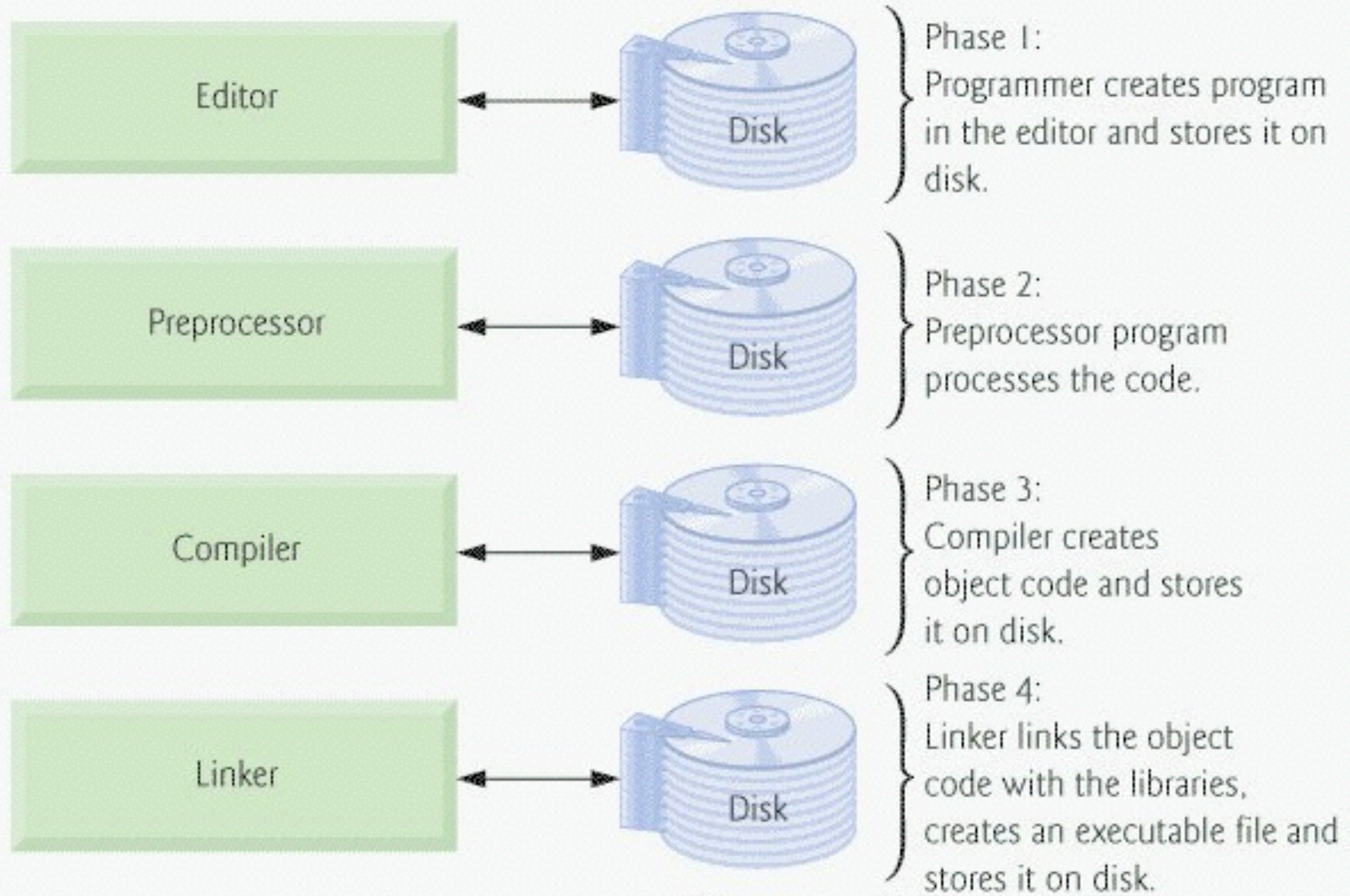
int main(void)
{
    char name[150];
    cout << "Hola Mundo!" << endl;
    cin >> name;
    cout << name;

    return 0;
}
```

1:-- readingString-V0.cpp All (8,17) (C++ /I Abbrev)

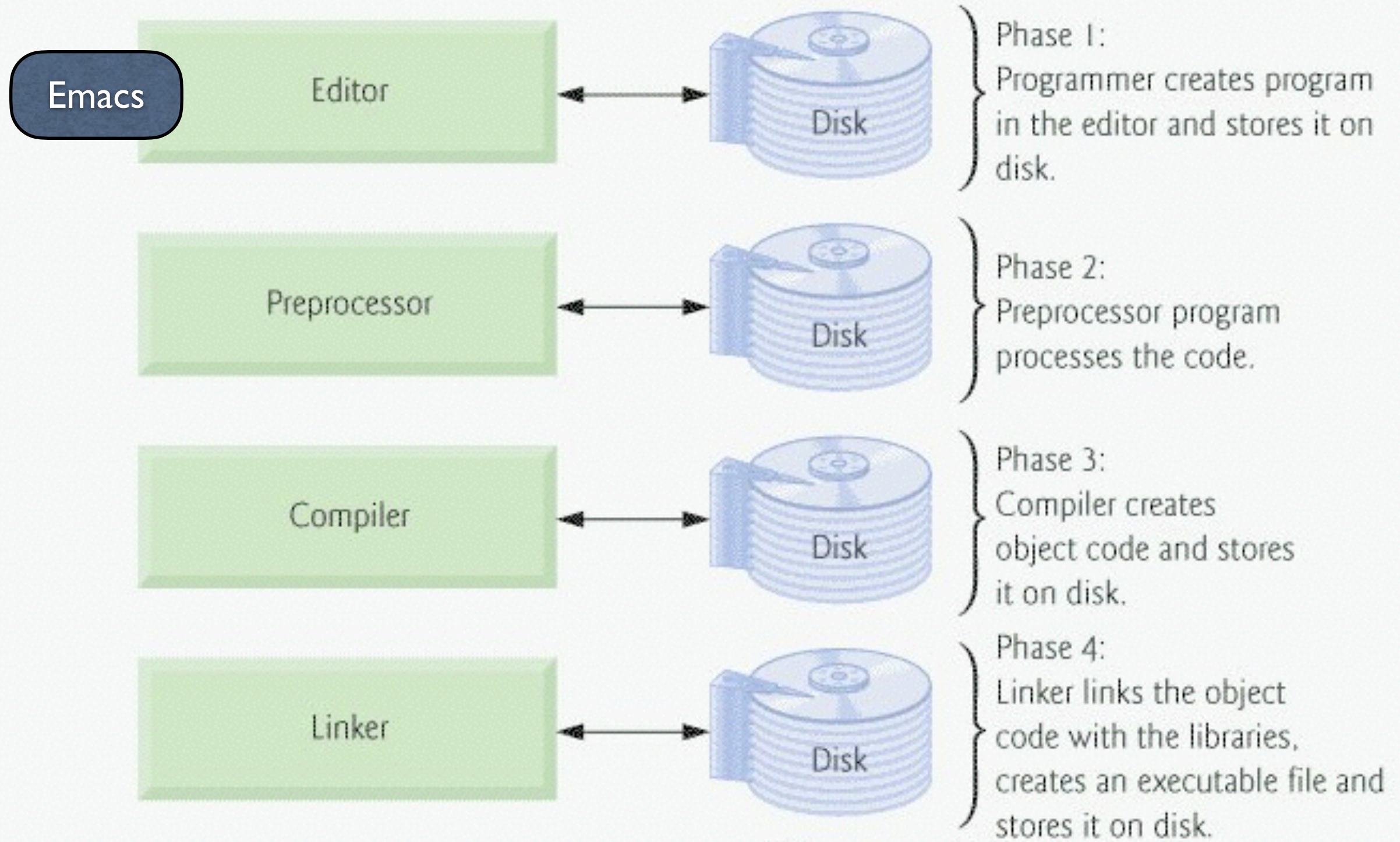
- Print your name.
- Print your age.

# C++ development cycle



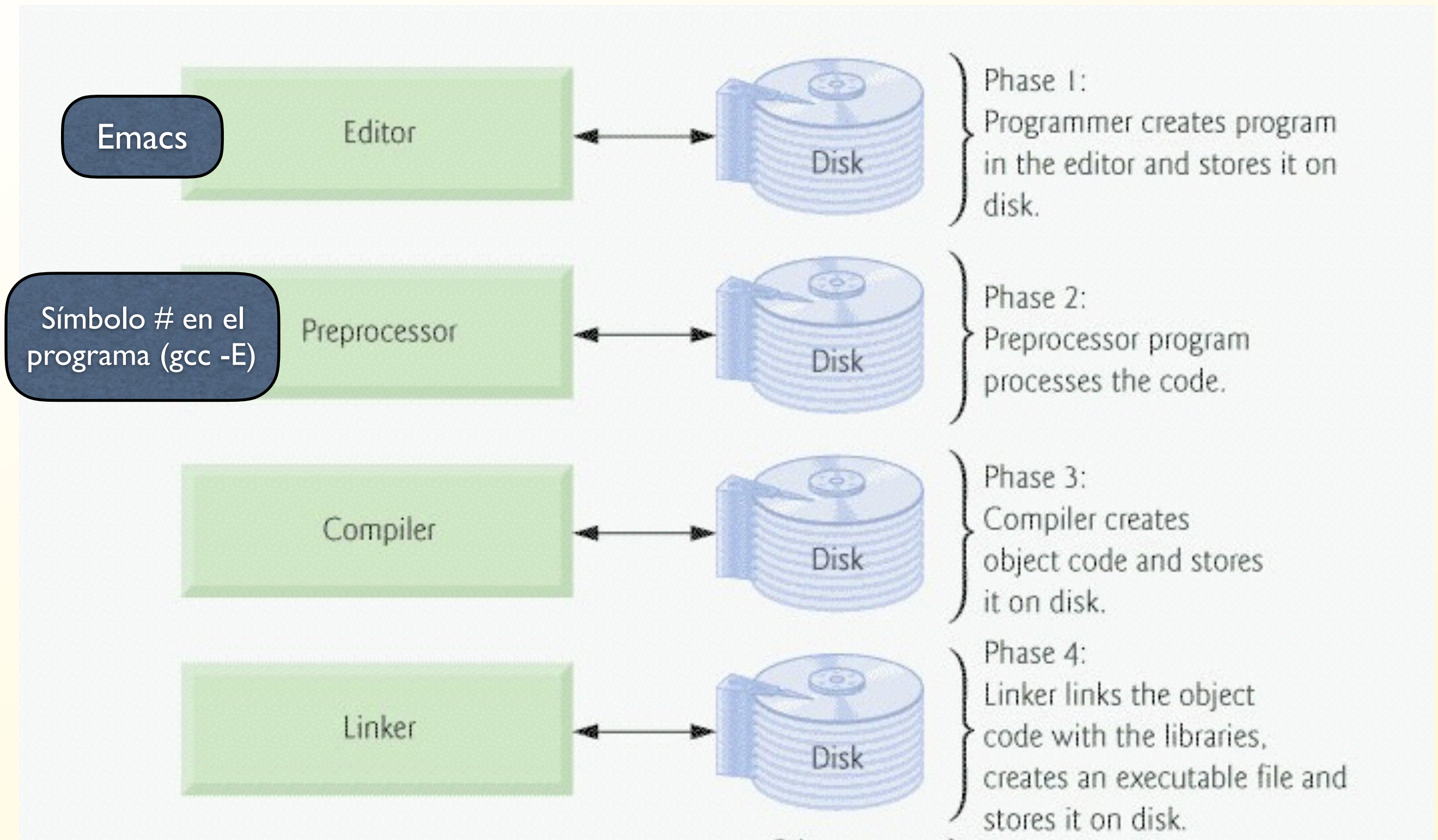


# C++ development cycle



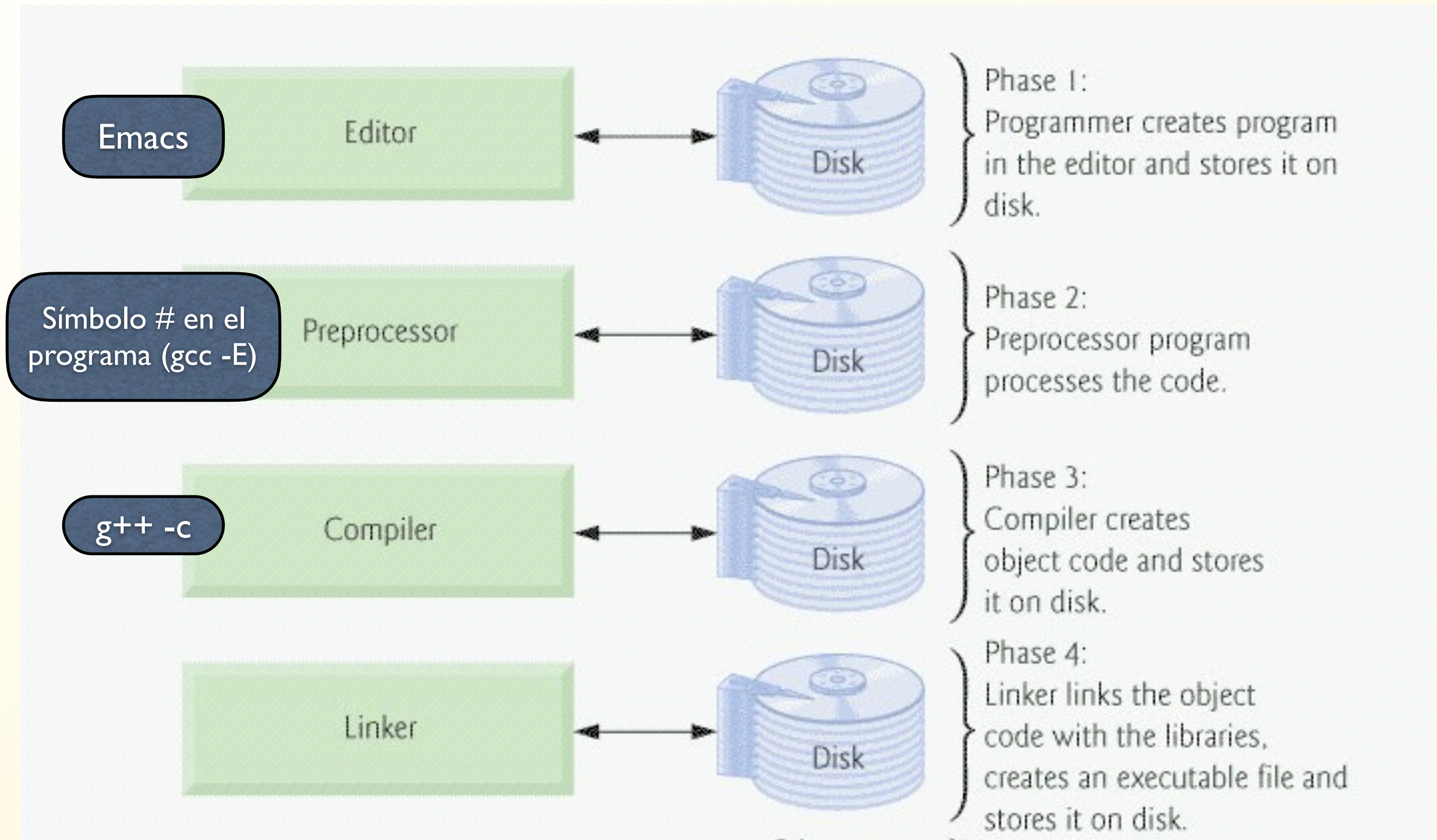


# C++ development cycle



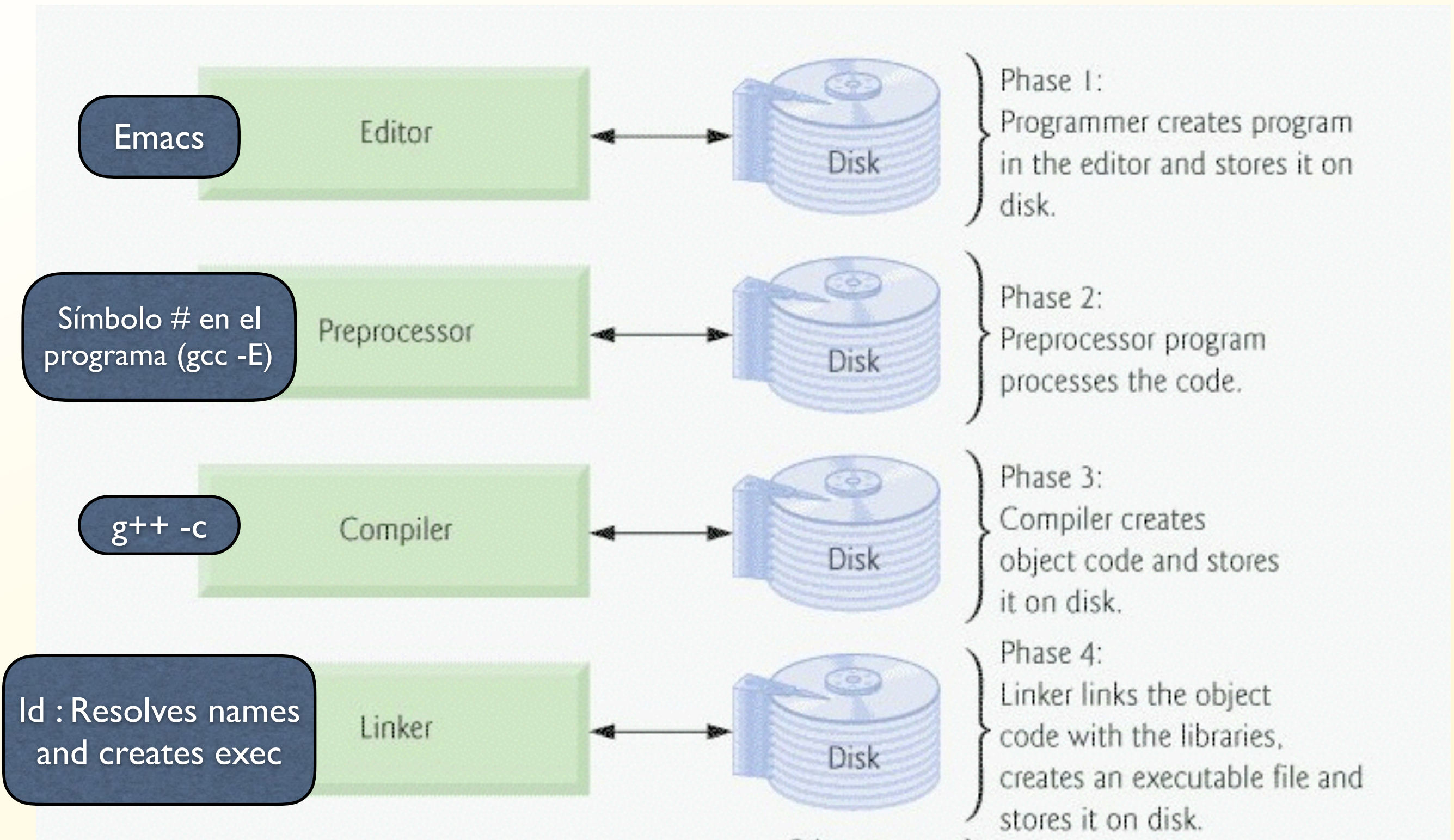


# C++ development cycle



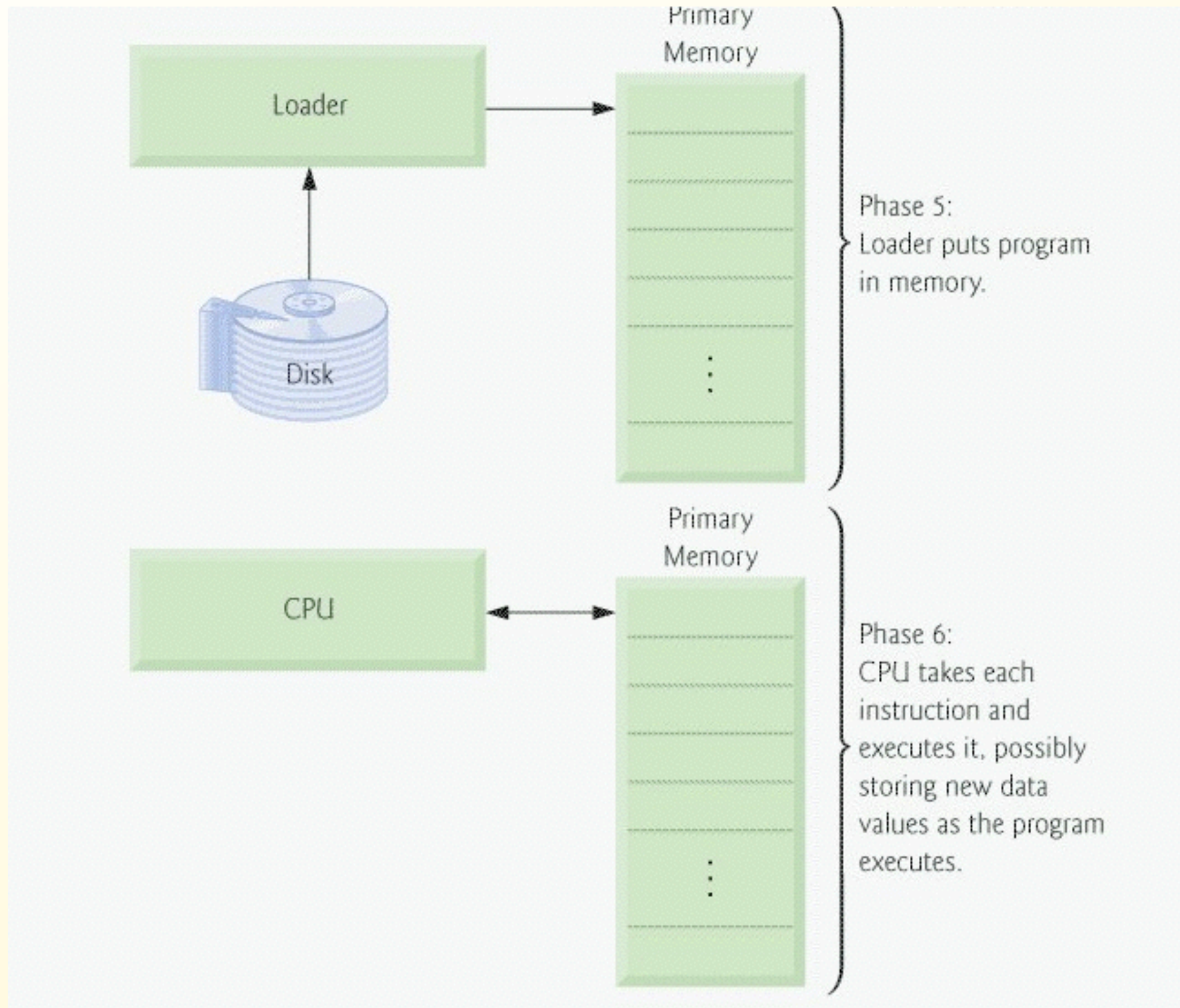


# C++ development cycle



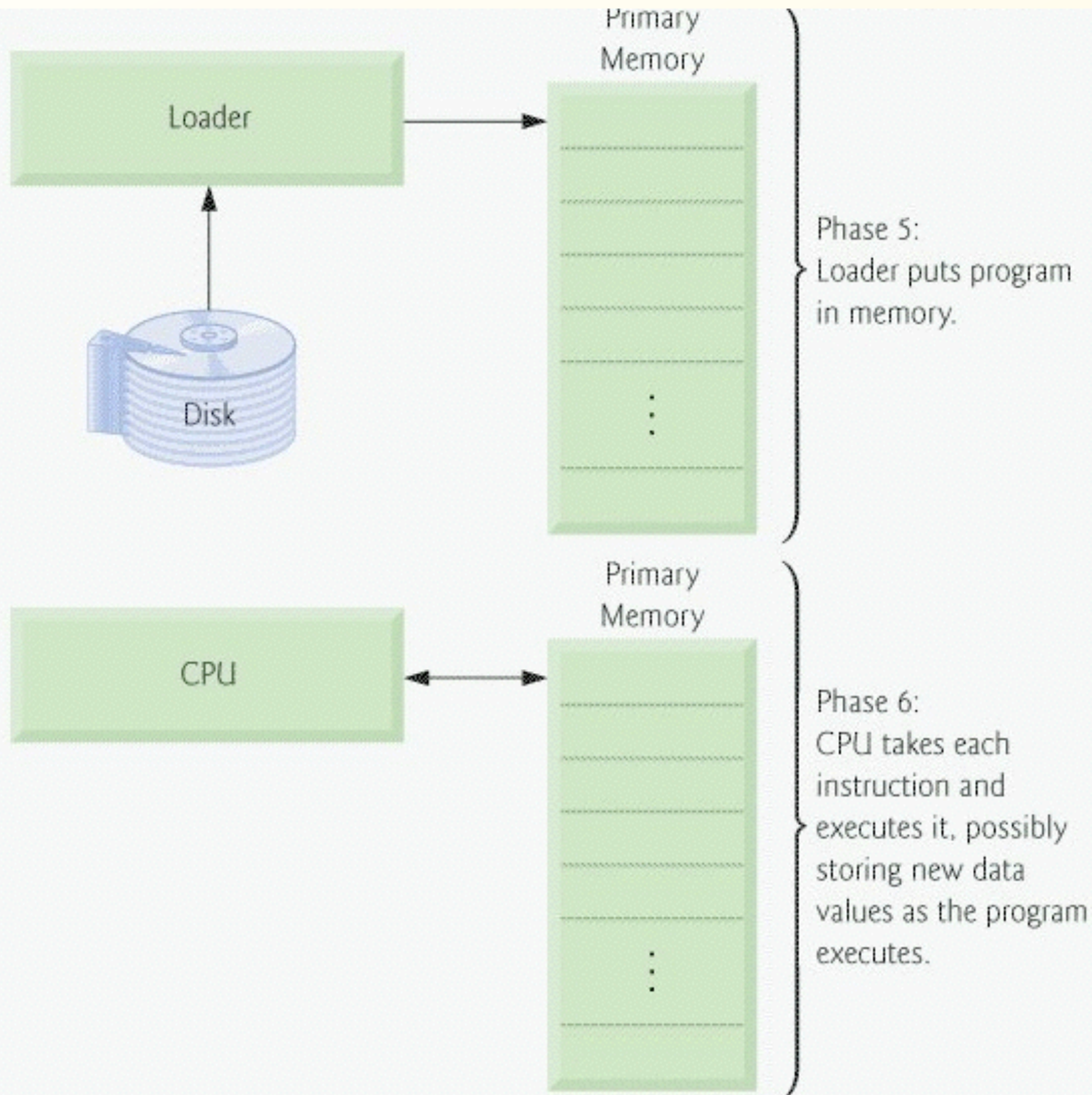


# C++ development cycle



# C++ development cycle

`./a.out`



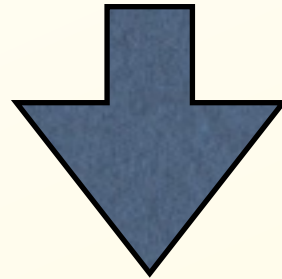
# Standard output, input, error

- Code 0 : Standard input (typically the keyboard)
- Code 1 : Standard output (typically the screen)
- Code 2 : Standar error (typically the screen)



# Standard output, input, error

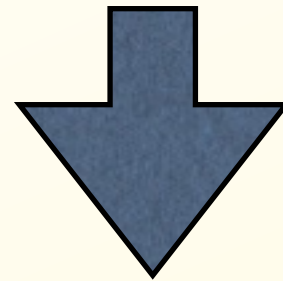
- Code 0 : Standard input (typically the keyboard)
- Code 1 : Standard output (typically the screen)
- Code 2 : Standar error (typically the screen)



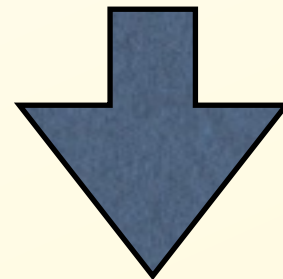
- Code 0 : cin (stdin)
- Code 1 : cout (stdout)
- Code 2 : cerr , clog (stderr)

# Standard output, input, error

- Code 0 : Standard input (typically the keyboard)
- Code 1 : Standard output (typically the screen)
- Code 2 : Standar error (typically the screen)



- Code 0 : cin (stdin)
- Code 1 : cout (stdout)
- Code 2 : cerr , clog (stderr)



- `./a.out > datos.dat` (*imprimir stdout al archivo datos.dat*)
- `./a.out |> datos.dat` (*imprimir stdout al archivo datos.dat*)
- `./a.out |> datos.dat 2>log.dat` (*imprimir stdout a datos.dat y el stderr a log.dat*)
- ...

# Basic program

```
1 // my first program in C++
2 #include <iostream>
3
4 int main()
5 {
6     std::cout << "Hello World!";
7 }
```



# Basic program

```
1 // my first program in C++
2 #include <iostream>
3
4 int main()
5 {
6     std::cout << "Hello World!";
7 }
```

```
1 int main ()
2 {
3     std::cout << " Hello World!";
4 }
```

# Basic program

```
1 // my first program in C++
2 #include <iostream>
3
4 int main()
5 {
6     std::cout << "Hello World!";
7 }
```

```
1 int main ()
2 {
3     std::cout << " Hello World!";
4 }
```

```
int main () { std::cout << "Hello World!"; }
```

# Basic program

```
1 // my first program in C++
2 #include <iostream>
3
4 int main()
5 {
6     std::cout << "Hello World!";
7 }
```

```
1 int main ()
2 {
3     std::cout << " Hello World!";
4 }
```

```
int main () { std::cout << "Hello World!"; }
```

```
1 int main ()
2 {
3     std::cout <<
4     "Hello World!";
5     std::cout
6     << "I'm a C++ program";
7 }
```



# Basic program

```
1 // my first program in C++
2 #include <iostream>
3
4 int main()
5 {
6     std::cout << "Hello World!";
7 }
```

```
1 int main ()
2 {
3     std::cout << " Hello World!";
4 }
```

```
int main () { std::cout << "Hello World!"; }
```

```
1 int main ()
2 {
3     std::cout <<
4         "Hello World!";
5     std::cout
6         << "I'm a C++ program";
7 }
```

```
int main()
{ // prints "Hello, World!":
    cout << "Hello, W" << 'o' << "rld" << '!' << '\n';
}
```

# Comments

```
1  /* my second program in C++  
2     with more comments */  
3  
4  #include <iostream>  
5  
6  int main ()  
7  {  
8     std::cout << "Hello World! ";    // prints Hello World!  
9     std::cout << "I'm a C++ program"; // prints I'm a C++ program  
0  }
```

# Ejercicios

1. Write a C++ program that displays your name and address (or if you value your privacy, a fictitious name and address).
2. Write a C++ program that asks for a distance in furlongs and converts it to yards.  
(One furlong is 220 yards.)

1. What is the correct value to return to the operating system upon the successful completion of a program?

- A. -1
- B. 1
- C. 0
- D. Programs do not return a value.

2. What is the only function all C++ programs must contain?

- A. `start()`
- B. `system()`
- C. `main()`
- D. `program()`

3. What punctuation is used to signal the beginning and end of code blocks?

- A. { }
- B. `->` and `<-`
- C. `BEGIN` and `END`
- D. ( and )

4. What punctuation ends most lines of C++ code?

- A. .
- B. ;
- C. :
- D. '

5. Which of the following is a correct comment?

- A. `*/ Comments */`
- B. `** Comment **`
- C. `/* Comment */`
- D. `{ Comment }`

6. What header file do you need to use to get access to `cout`?

- A. `stream`
- B. nothing, it is available by default
- C. `iostream`
- D. `using namespace std;`



**1.1** Describe the two ways to include comments in a C++ program.

**1.2** What is wrong with this program?

```
#include <iostream>
int main()
{ // prints "Hello, World!":
  cout << "Hello, World!\n"
}
```

**1.3** What is wrong with the following C-style comment?

```
cout << "Hello, /* change? */ World.\n";
```

**1.4** What's wrong with this program:

```
#include <iostream>;
int main
{ // prints "n = 22":
  n = 22;
  cout << "n = << n << endl;
}
```

**1.5** What does a declaration do?

**1.6** What is the purpose of the preprocessing directive:

```
#include <iostream>
```

**1.7** What is the shortest possible C++ program?

**1.8** Where does the name “C++” come from?

**1.9** What's wrong with these declarations:

```
int first = 22, last = 99, new = 44, old = 66;
```

**1.10** In each of the following, assume that `m` has the value 5 and `n` has the value 2 before the statement executes. Tell what the values of `m` and `n` will be after each of the following statements executes:

*a.* `m *= n++;`

*b.* `m += --n;`

**1.11** Evaluate each of the following expressions, assuming in each case that `m` has the value 25 and `n` has the value 7:

*a.* `m - 8 - n`

*b.* `m = n = 3`

*c.* `m%n`

*d.* `m%n++`

*e.* `m%++n`

*f.* `++m - n--`

- 1.1** Write four different C++ statements, each subtracting 1 from the integer variable `n`.
- 1.2** Write a block of C++ code that has the same effect as the statement
- ```
n = 100 + m++;
```
- without using the post-increment operator.
- 1.3** Write a block of C++ code that has the same effect as the statement
- ```
n = 100 + ++m;
```
- without using the pre-increment operator.
- 1.4** Write a single C++ statement that subtracts the sum of `x` and `y` from `z` and then increments `y`.
- 1.5** Write a single C++ statement that decrements the variable `n` and then adds it to `total`.
- 1.6** Write a program that prints the first sentence of the Gettysburg Address (or your favorite quotation).
- 1.7** Write a program that prints the block letter “B” in a  $7 \times 6$  grid of stars like this:
- ```
*****
*       *
*       *
*****
*       *
*       *
*****
```
- 1.8** Write and run a program that prints the first letter of your last name as a block letter in a  $7 \times 7$  grid of stars.
- 1.9** Write and run a program that shows what happens when each of the following ten “escape sequences” is printed: `\a`, `\b`, `\n`, `\r`, `\t`, `\v`, `\'`, `\"`, `\\`, `\?`.



- 1.12** Parse the following program, identifying all the keywords, identifiers, operators, literals, punctuation, and comments:

```
int main()
{ int n;
  cin >> n;
  n *= 3;  // multiply n by 3
  cout << "n=" << n << endl;
}
```

- 1.13** Identify and correct the error in each of the following:

*a.* `cout >> count;`

*b.* `int double=44;`

- 1.14** How do the following two statements differ:

`char ch = 'A';`

`char ch = 65;`

- 1.15** What code could you execute to find the character whose ASCII code is 100?

- 1.16** What does “floating-point” mean, and why is it called that?

- 1.17** What is numeric overflow?

- 1.18** How is integer overflow different from floating-point overflow?

- 1.19** What is a run-time error? Give examples of two different kinds of run-time errors.

- 1.20** What is a compile-time error? Give examples of two different kinds of compile-time errors.