



**NOTA RECORDATORIA.** La evaluación final consta de las siguientes pruebas:

Prueba escrita → 60% de la nota (mínimo 5 para aprobar)

Prácticas → 40% de la nota

## Ejercicio 1

[3 puntos]

El siguiente programa en C++ se compila con el comando `g++ main.cc`. Ignora errores tipográficos y potenciales memory leaks. ¿Cómo se comportaría al compilar? ¿Por qué no compilaría? Si compila, ¿qué sacaría por pantalla?

### Programa A - C++

foo.h	bar.h	main.cc
<pre>class Foo { protected:     int i; public:     Foo(int i_):         i(i_) {}      virtual void sil()     { i*=2; }      int meta() const     { return i; } };</pre>	<pre>#include "foo.h" class Bar: public Foo { public:     Bar() :         Foo(3) { }      void sil() override     { i+=2; }      int meta() const     { return i-1; } };</pre>	<pre>#include &lt;iostream&gt; #include "bar.h" using namespace std;  void mira(Foo* foo) {     foo-&gt;sil();     cout&lt;&lt;foo-&gt;meta();     cout&lt;&lt;endl; }  int main() {     mira(new Foo(1));     mira(new Bar()); }</pre>

### Programa B - C++

foo.h	bar.h	main.cc
<pre>class Foo { public:     int mitzvah() const     { return 1; } };</pre>	<pre>template &lt;typename T&gt; class Bar {     T t; public:     Bar(const T&amp; t_) :         t(t_+1) {}      int mitzvah() const {         return t.mitzvah()*2;     } };  template&lt;typename T&gt; Bar&lt;T&gt; bar(const T&amp; t) {     return Bar&lt;T&gt;(t); }</pre>	<pre>#include "bar.h" #include "foo.h" #include &lt;iostream&gt; using namespace std;  template&lt;typename T&gt; void show(const T&amp; t) {     cout&lt;&lt;t.mitzvah()&lt;&lt;endl; }  int main() {     show(Foo());     show(bar(Foo()));     show(bar(bar(Foo())));     show(bar(bar(bar(Foo())))); }</pre>

## Programa C - Haskell

Foo.hs	Main.hs
<pre>module Foo where foo :: (Ord a) =&gt; [a] -&gt; Int foo [] = 0 foo [x] = 1 foo (x:y:xs)     x &gt; y = 1 + foo (y:xs)     otherwise = foo (y:xs)</pre>	<pre>import Foo  main = do   print \$ foo [1,2,3,1,2,3]   print \$ foo "abcbcdcde"   print \$ foo [1,3..21]</pre>