Sobrecarga: operadores de comparación (==, !=)

Operadores de comparación: ejemplo

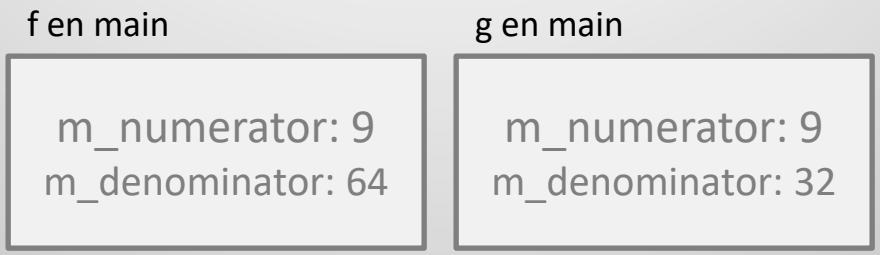
```
// main.cpp
...
int main()
{
    if(f==g)
    ...

// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f==g)
     ...
```




```
// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ...f==g)
    ...
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ...f==g)
     ...
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f==g)
     ...

// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

9

64

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ...f==g)
    ...
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ...f==g)
    ...

// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator...rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f==g)
     ...

// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

9 32 576

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ... f==g)
    ...

// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

288

576

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ... f==g)
    ...

// fraction.h
...
class Fraction
{
    friend bool operator==(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator==(const Fraction & lhs, const Fraction & rhs)
{
    return lhs.m_numerator*rhs.m_denominator==rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator==

m_numerator: 9
m_denominator: 64

g en main
rhs en operator==

m_numerator: 9
m_denominator: 32

288

false

576

Operadores de comparación: ejemplo

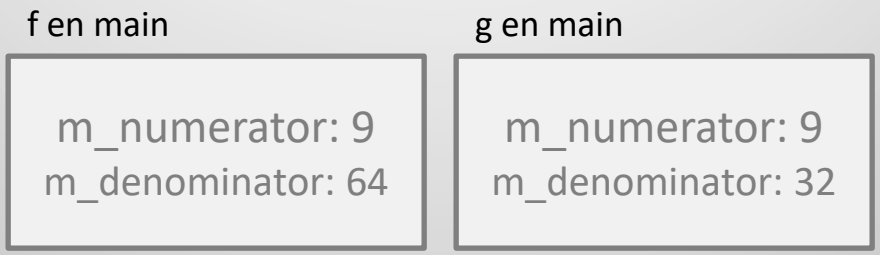
```
// main.cpp
...
int main()
{
    if(f!=g)
    ...

// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...
```




```
// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...

// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

9

64

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    ...f!=g)
     ...
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator...rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...

// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

9 32 576

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...

// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

288

576

Operadores de comparación: ejemplo

```
// main.cpp
...
int main()
{
    if(f!=g)
     ...
```

f en main
lhs en operator!=

m_numerator: 9
m_denominator: 64

g en main
rhs en operator!=

m_numerator: 9
m_denominator: 32

```
// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
    return lhs.m_numerator*rhs.m_denominator!=rhs.m_numerator*lhs.m_denominator;
}
```

288

true

576

Definición de != utilizando ==

```
// main.cpp
...
int main()
{
    if(f!=g)
    ...

// fraction.h
...
class Fraction
{
    friend bool operator!=(const Fraction & lhs, const Fraction & rhs);
    ...

// fraction.cpp
bool operator!=(const Fraction & lhs, const Fraction & rhs)
{
  return !(lhs==rhs);
}
```