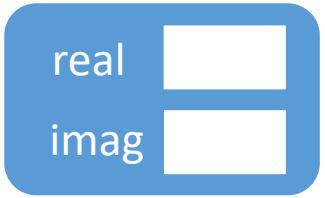
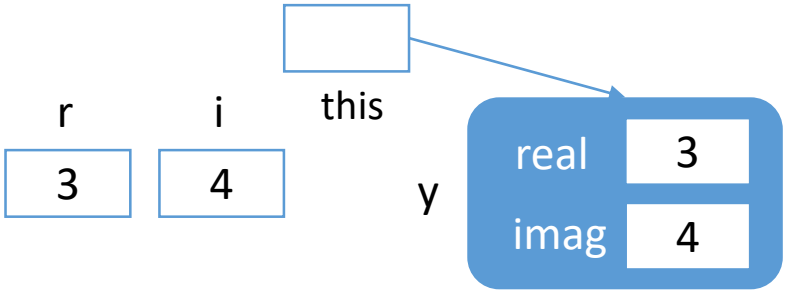
Step-by-step walkthrough for example on page 34 – 35 of the lecture notes:
Data Abstraction & Classes

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

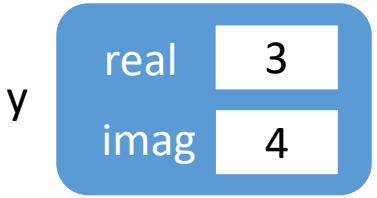
y

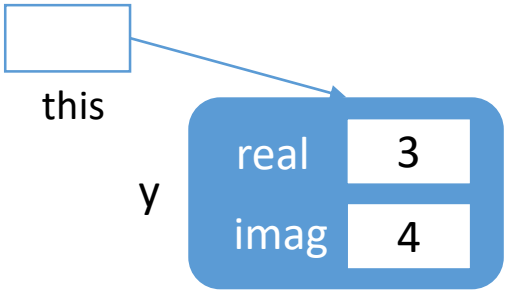





```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```






```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

Output

```
( 3 , 4 )
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

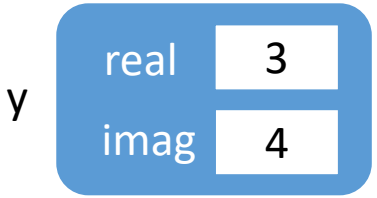
y

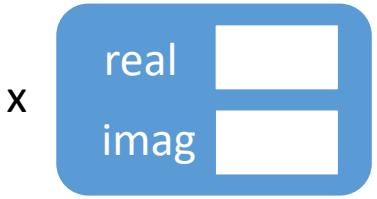
real	3
imag	4

Output

```
( 3 , 4 )

Return by value
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

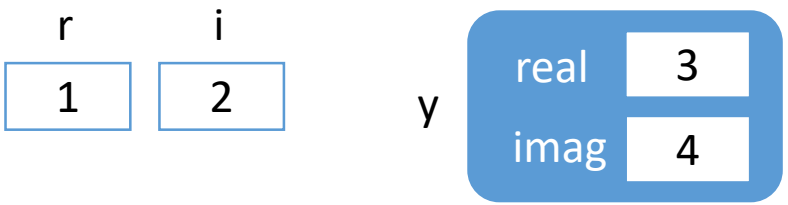


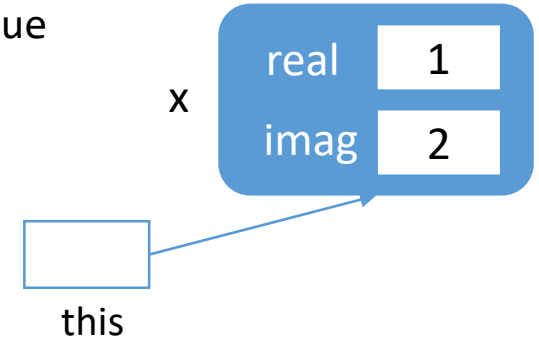


Output

```
( 3 , 4 )

Return by value
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

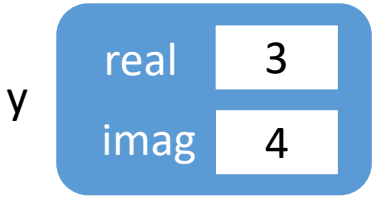




Output

```
( 3 , 4 )

Return by value
```

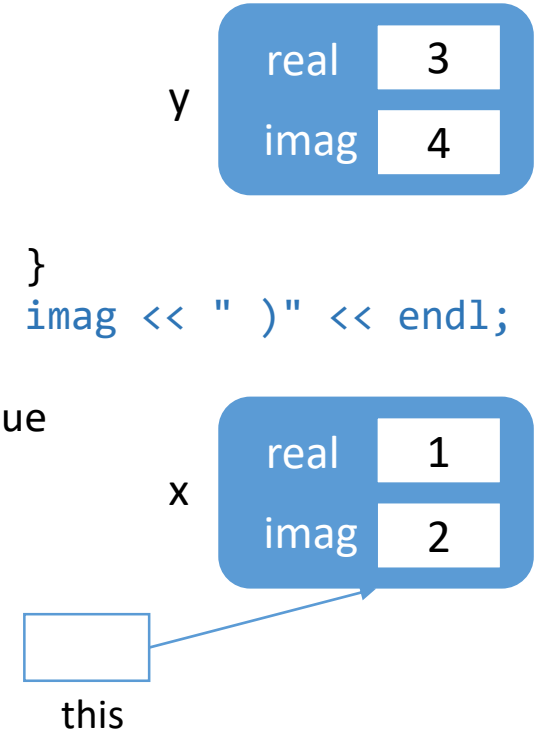
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```



Output

```
( 3 , 4 )

Return by value
```

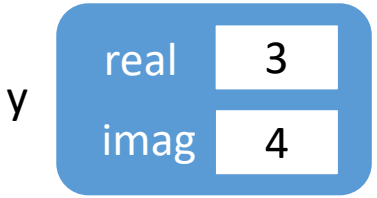
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

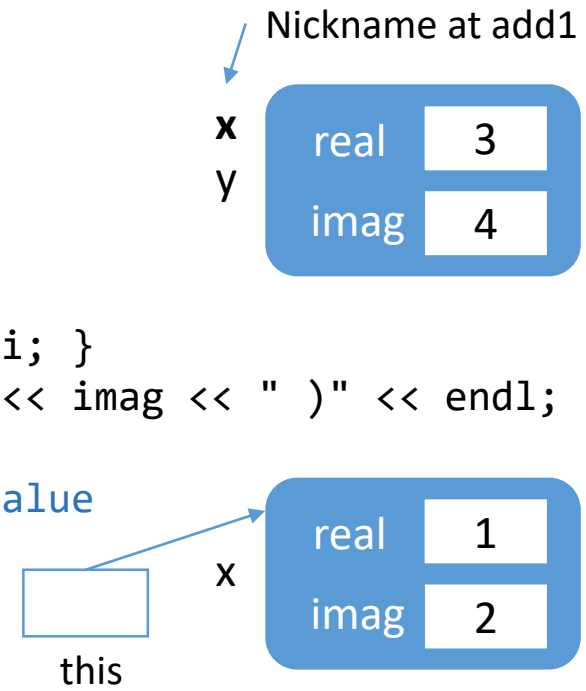
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

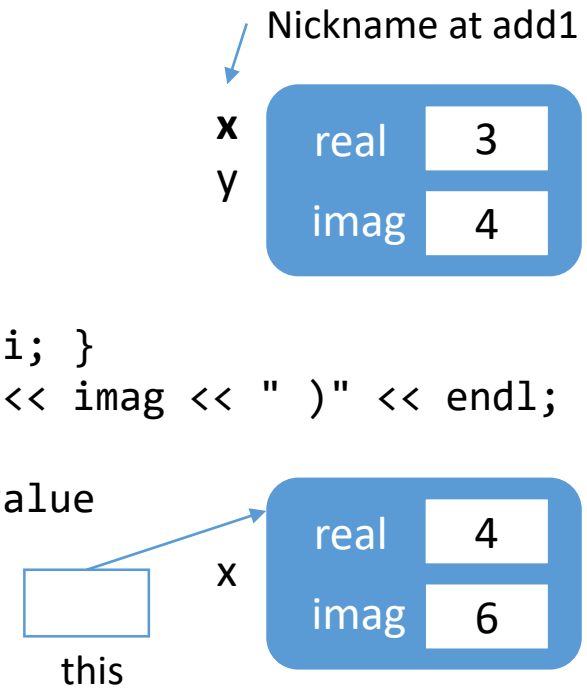
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

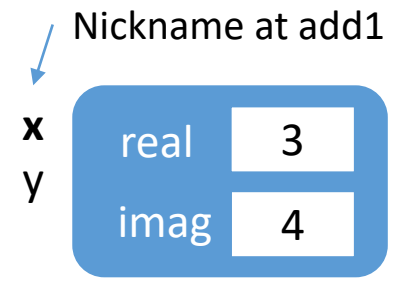
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

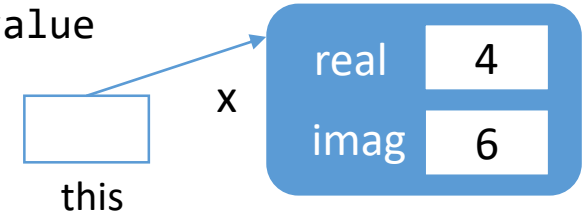


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

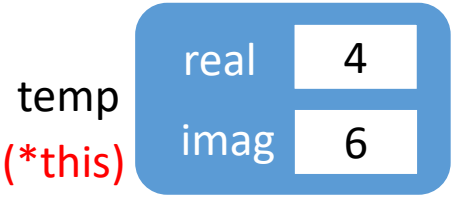
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```





Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```



Generated by return (*this)

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
temp ~~x.add1(y)~~.add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

y

real	3
imag	4

x

real	4
imag	6

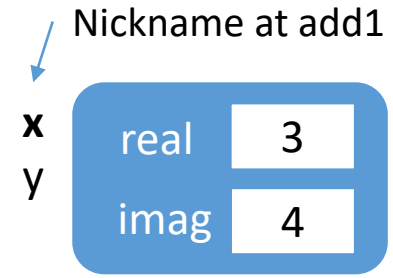
Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

temp

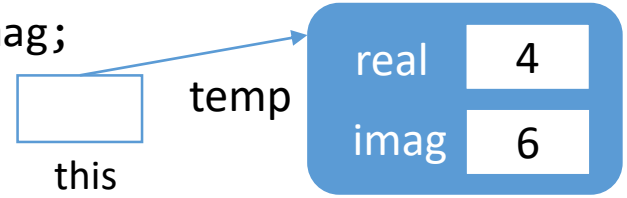
real	4
imag	6

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

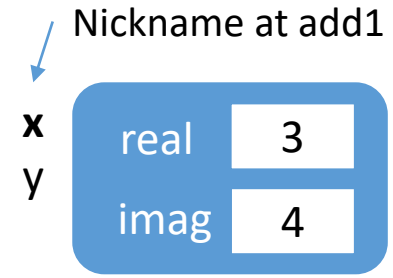


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

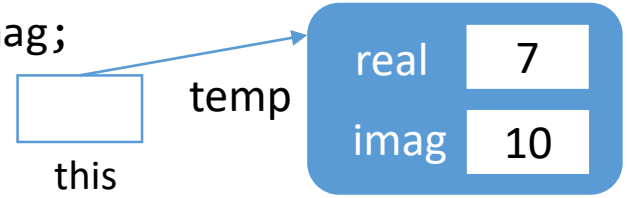


```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

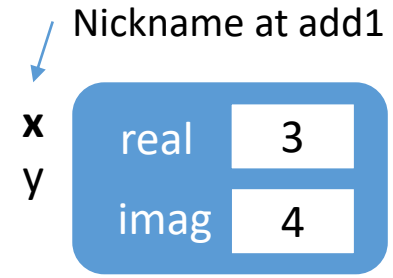


Output

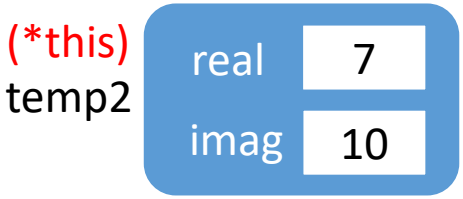
```
( 3 , 4 )

Return by value
( 1 , 2 )
```



```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

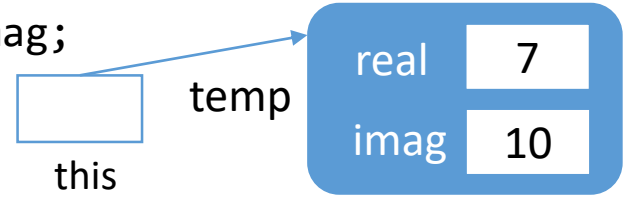


Generated by return (*this)



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```



```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

temp2

y

real	3
imag	4

x

real	4
imag	6

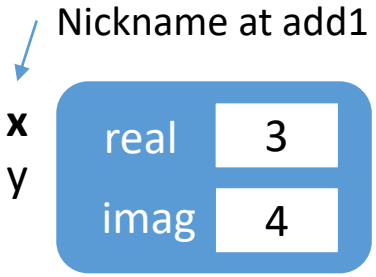
temp2

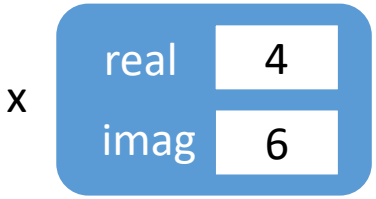
real	7
imag	10

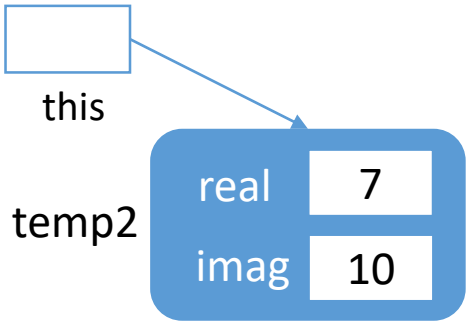
Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```







Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

y

real	3
imag	4

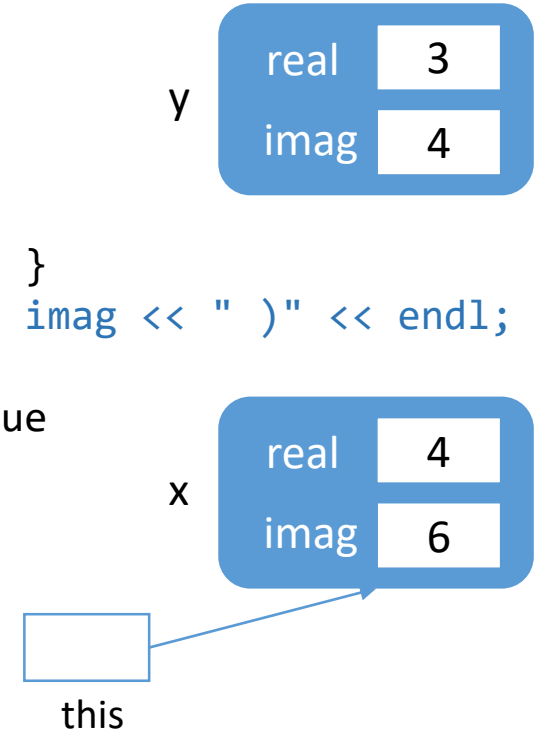
x

real	4
imag	6

Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
```

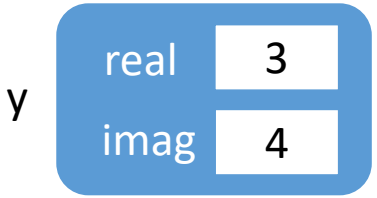
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

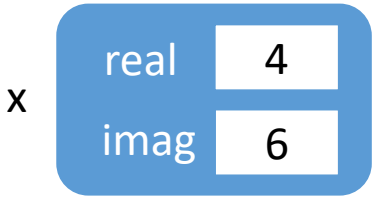


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

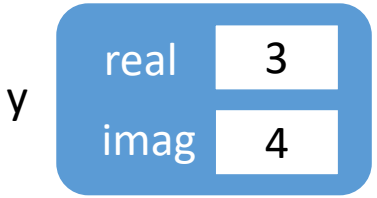


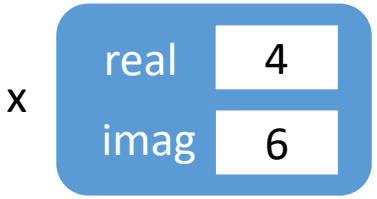


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

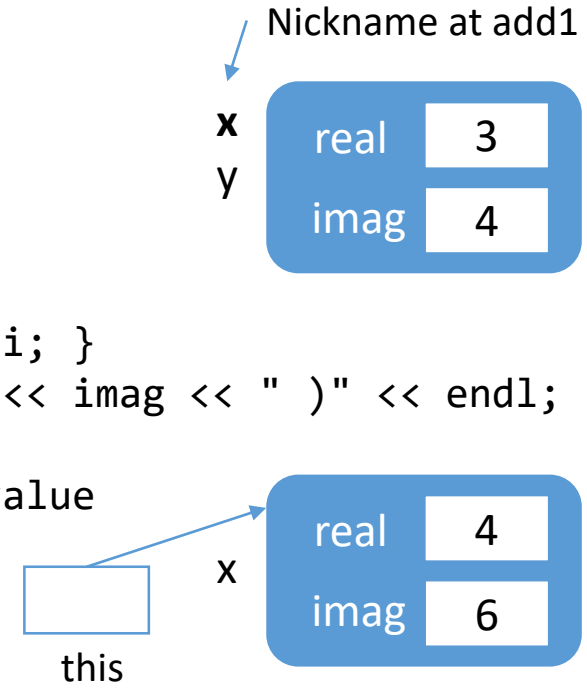




Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

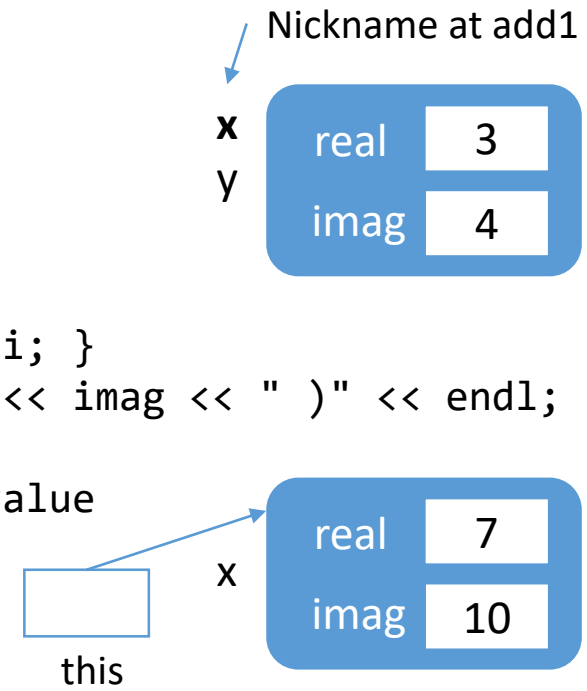
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

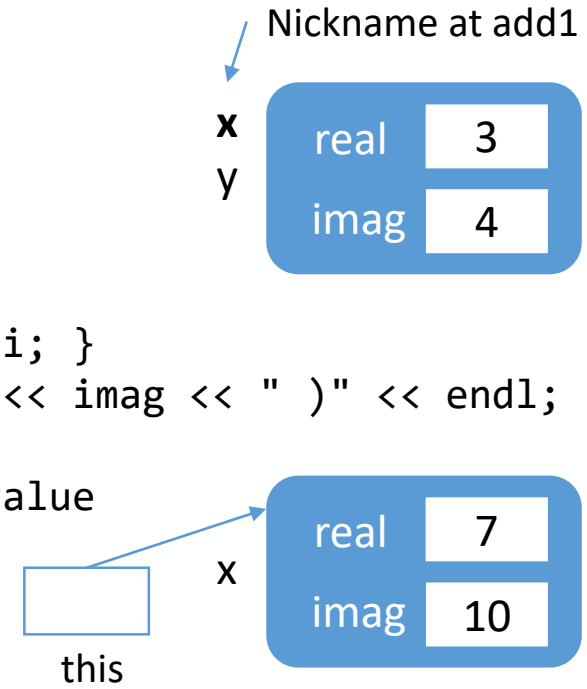
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

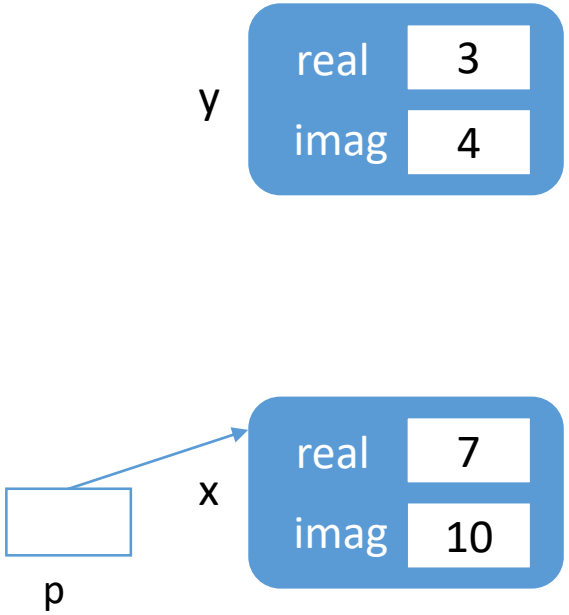
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

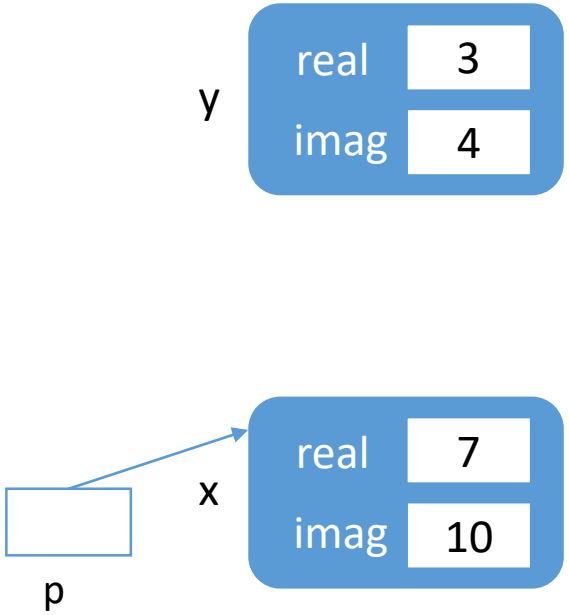
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

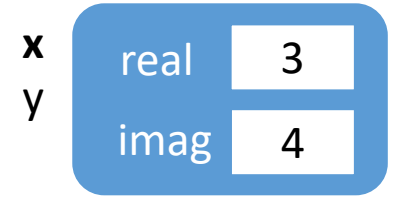
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

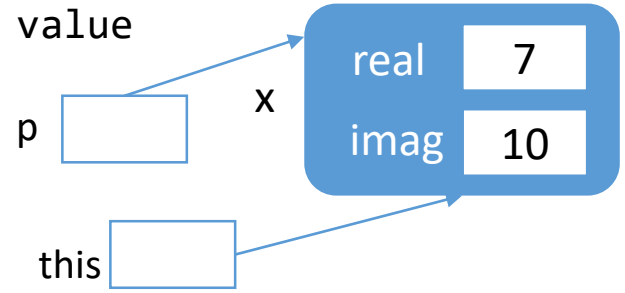


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

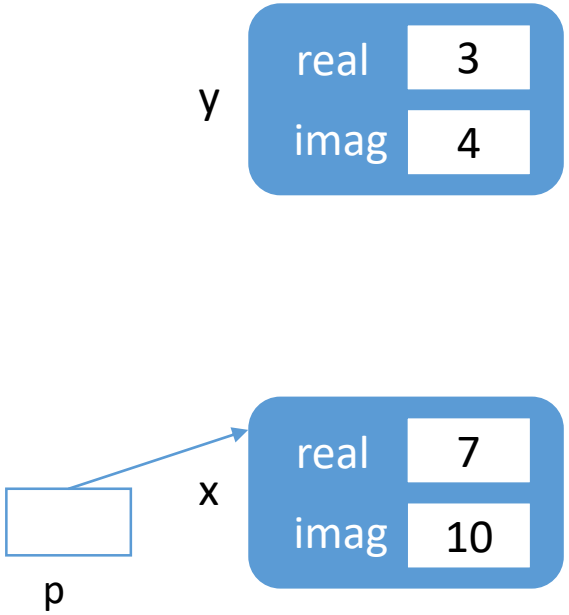




Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )
```

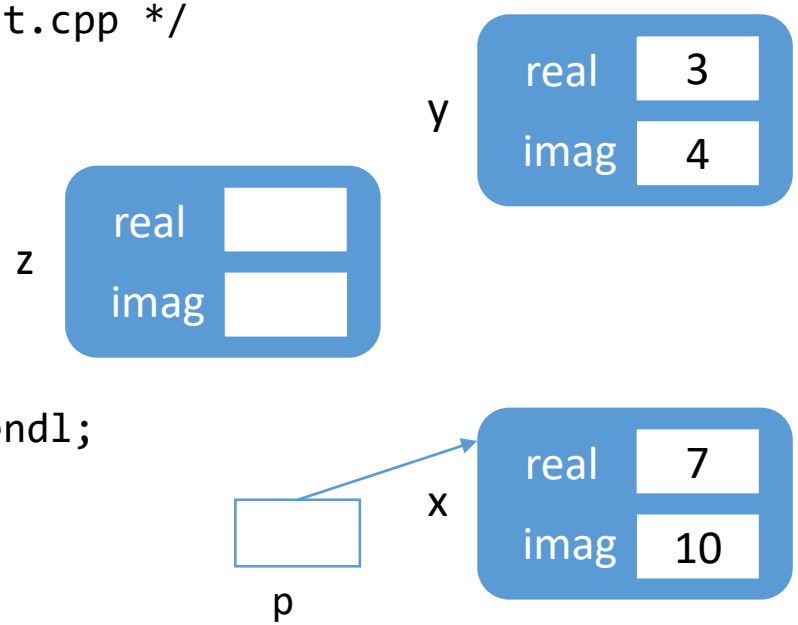
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
```

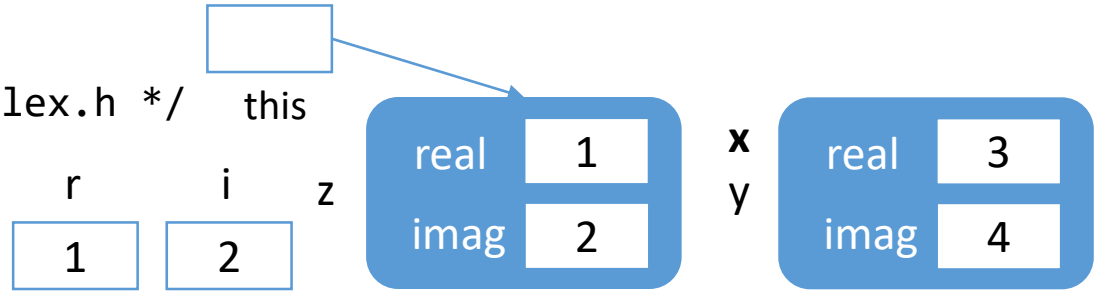
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

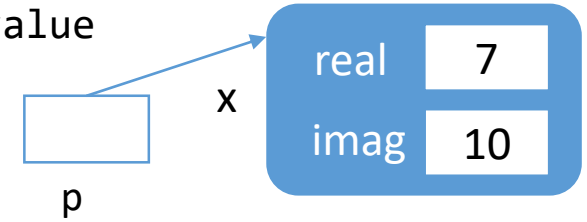


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

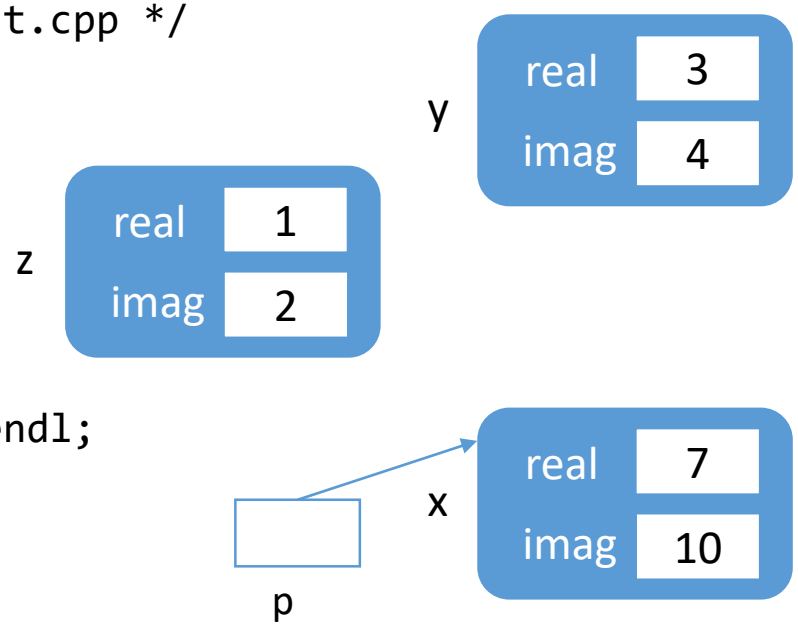




Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
```

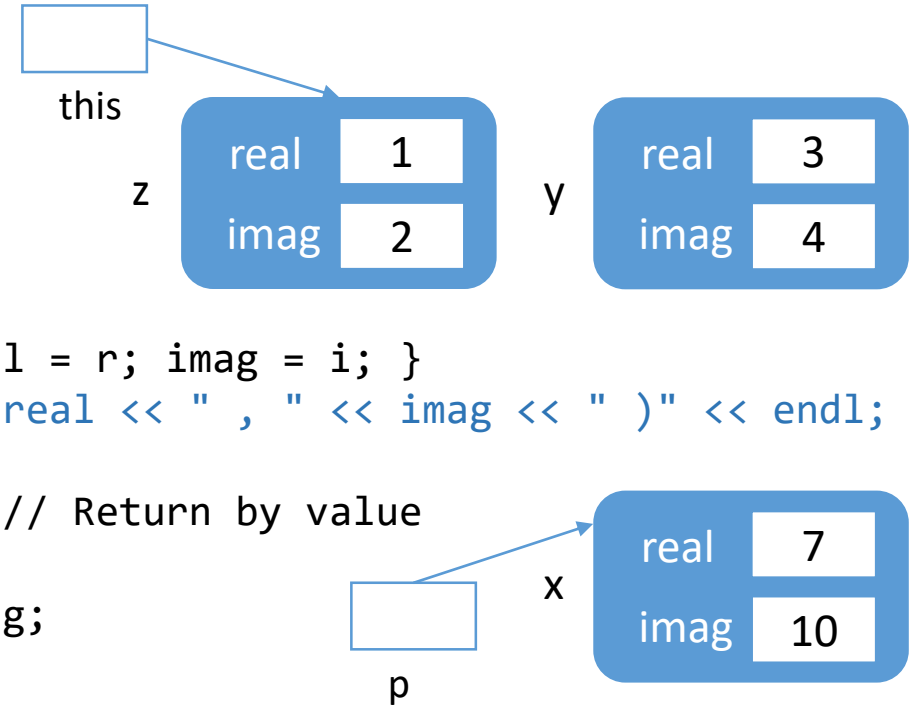
```cpp
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
```

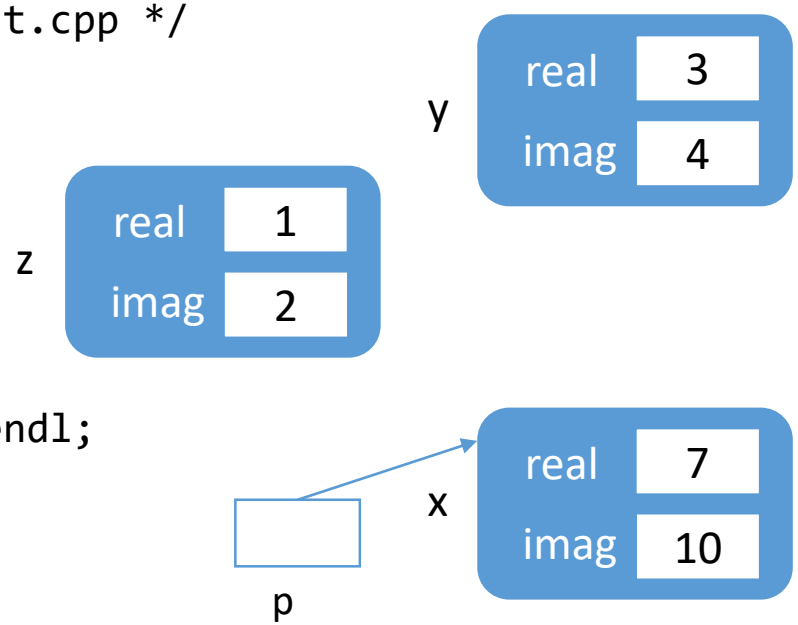
```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

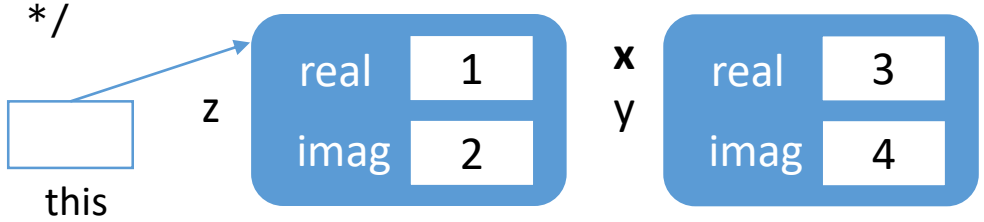
```cpp
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

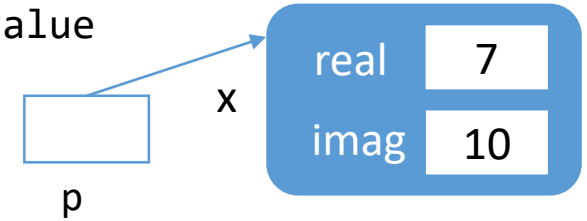


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

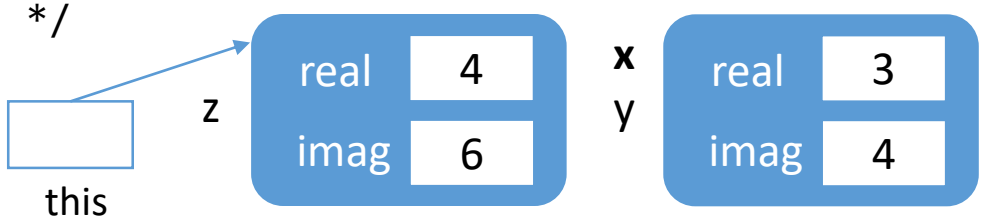


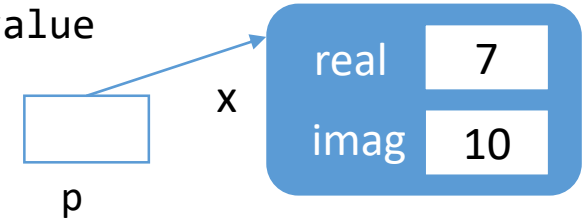


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

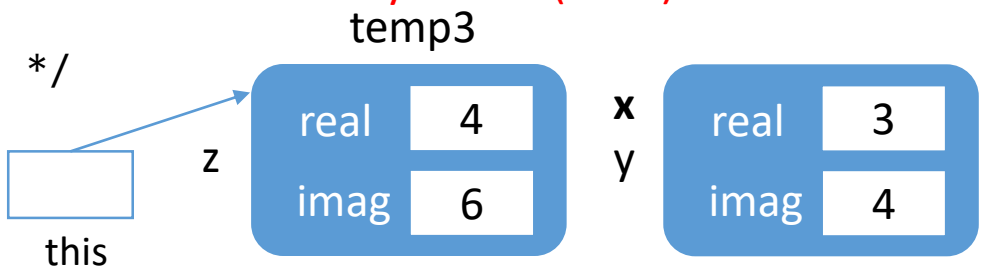




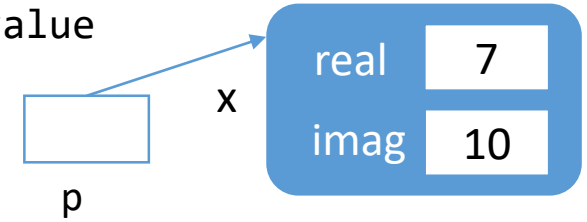
Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

Generated by return (*this)



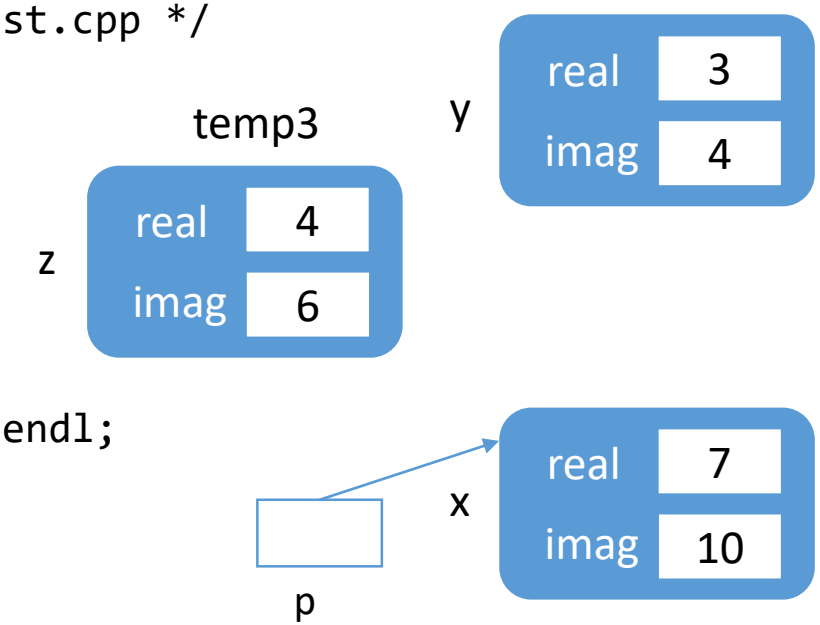

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

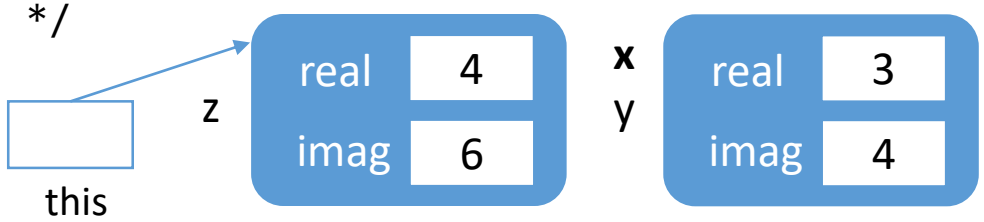
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
temp3 ~~z.add3(y)~~.add3(y).print();
  z.print();
  return 0;
}
```

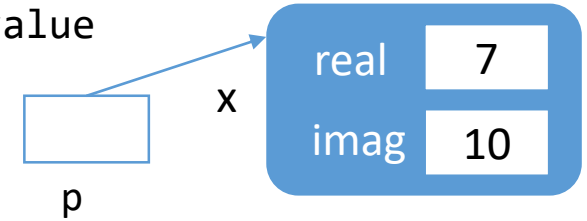


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

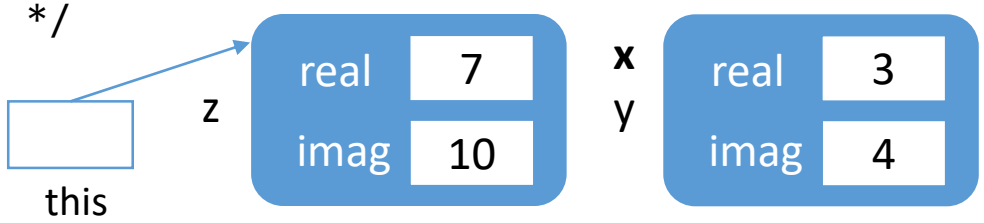


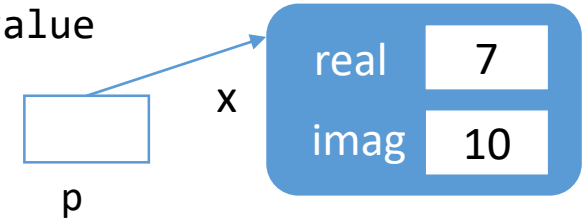


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

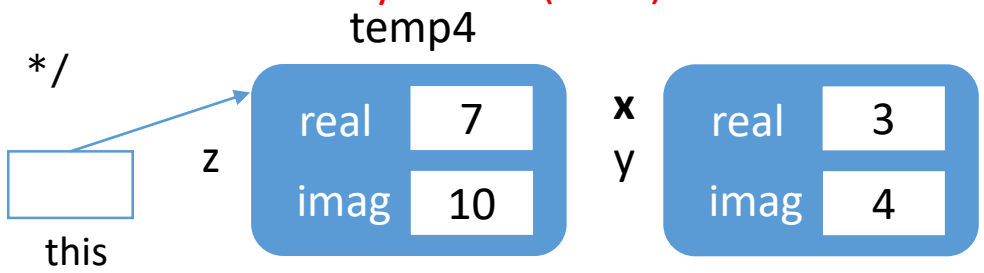




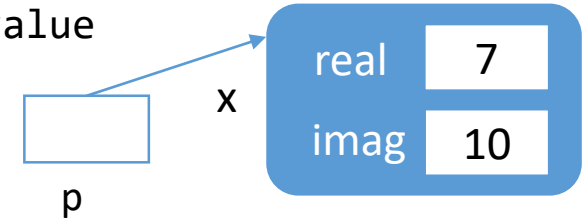
Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

Generated by return (*this)



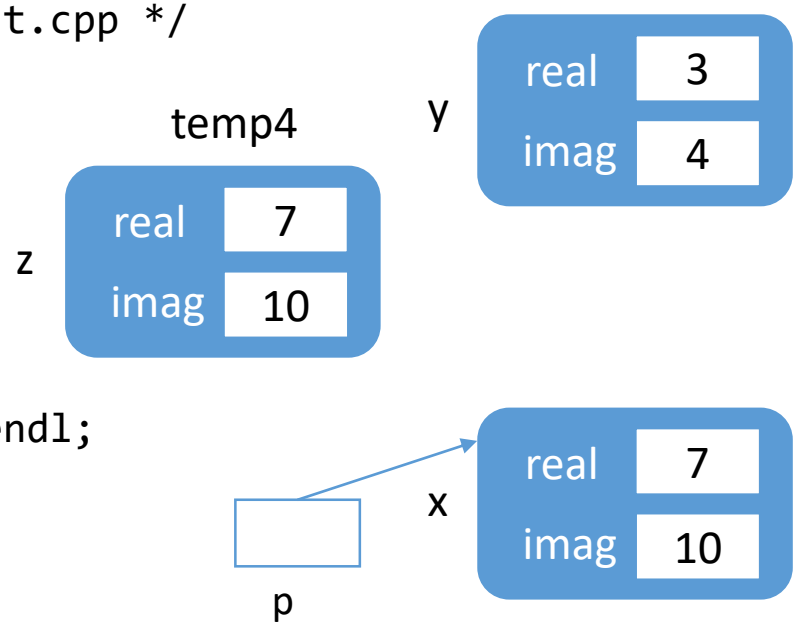

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

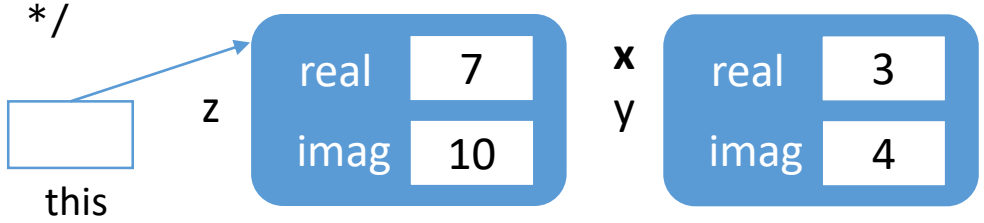
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
temp4 z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

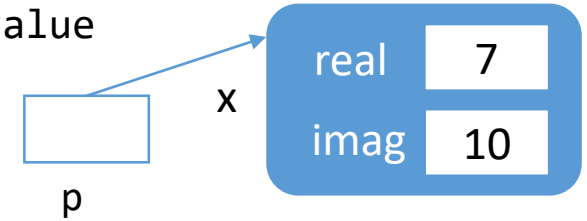


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

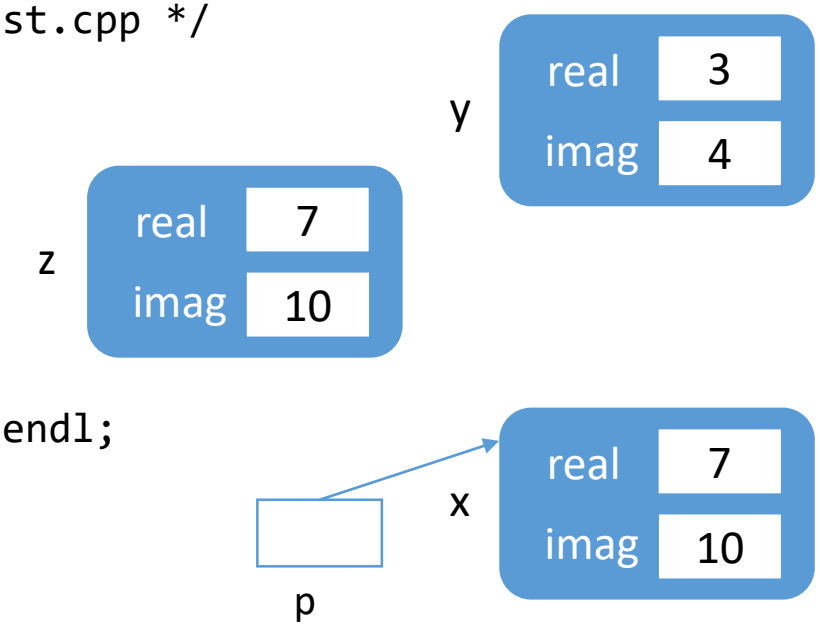




Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
( 7, 10 )
```

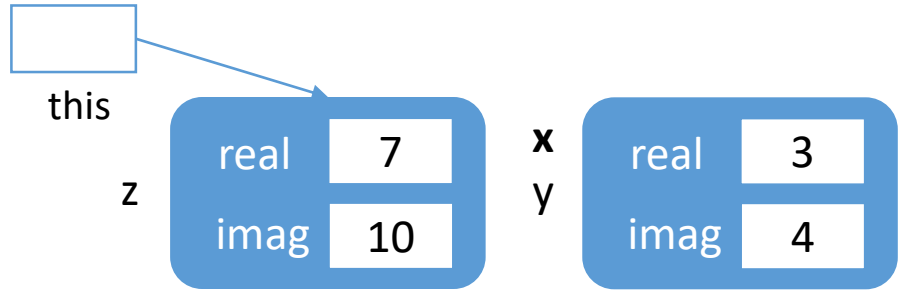
```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```

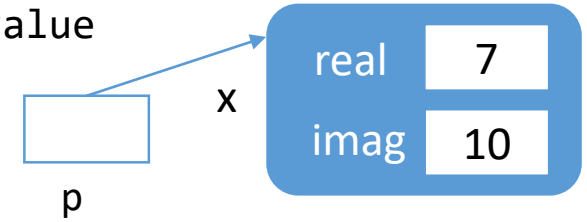


Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
( 7, 10 )
```

```
class Complex /* File: complex.h */
{
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }

    Complex add1(const Complex& x) // Return by value
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
    Complex* add2(const Complex& x)
    // Return by value using pointer
    {
      real += x.real; imag += x.imag;
      return this;
    }
    Complex& add3(const Complex& x)
    // Return by reference
    {
      real += x.real; imag += x.imag;
      return (*this);
    }
};
```

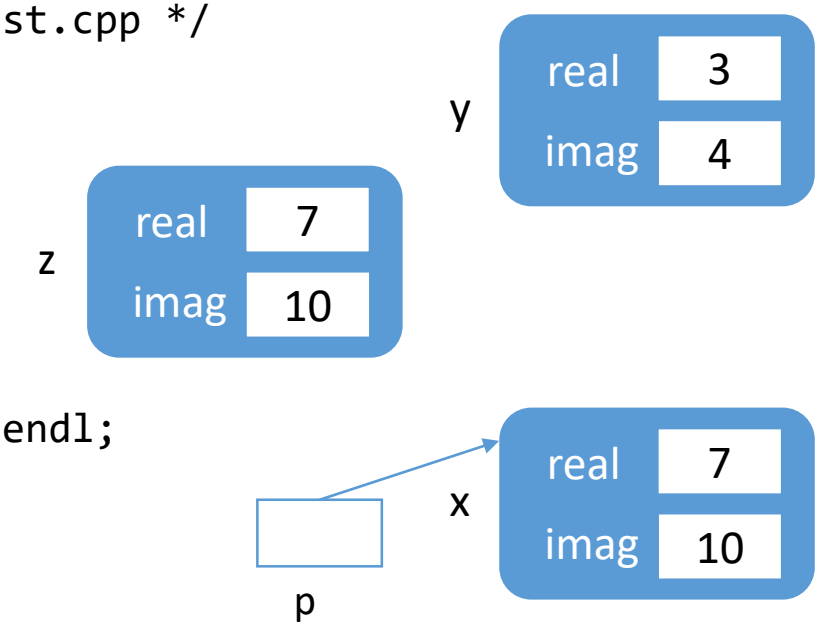




Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
( 7, 10 )
( 7, 10 )
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h“

int main()
{
  Complex y(3, 4);
  y.print();
  cout << endl << "Return by value" << endl;
  Complex x(1, 2);
  x.print();
  x.add1(y).add1(y).print();
  x.print();
  cout << endl << "Return its pointer by value" << endl;
  Complex* p = x.add2(y);
  p->print();
  cout << endl << "Return by reference" << endl;
  Complex z(1, 2);
  z.print();
  z.add3(y).add3(y).print();
  z.print();
  return 0;
}
```



Output

```
( 3 , 4 )

Return by value
( 1 , 2 )
( 7 , 10 )
( 4 , 6 )

Return its pointer by value
( 7 , 10 )

Return by reference
( 1 , 2 )
( 7, 10 )
( 7, 10 )
```