Step-by-step walkthrough for example on page 15 – 16 of the lecture notes:

Revision Example, Pointer,

Reference & const-ness

```
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; }</pre>
    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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
class Complex /* File: complex.h */
                                                       this
                                                  5
                                                           u
 private:
    float real; float imag;
 public:
    Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    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);
};
```

```
class Complex /* File: complex.h */
                                                       this
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &v
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
      real += x.real; imag += x.imag;
      return (*this);
                                                      this
    Complex* add2(const Complex& x) // Return by value using pointer
                                                             Output
      real += x.real; imag += x.imag;
                                                             (4,5)
      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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &v
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                           u
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;</pre>
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.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; }</pre>
    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
                                                             Output
      real += x.real; imag += x.imag;
                                                             (4,5)
      return this:
                                                             (4,5)
    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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.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; }</pre>
    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
                                                             Output
      real += x.real; imag += x.imag;
                                                             (4,5)
      return this;
                                                             (4,5)
                                                             (4,5)
    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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                           W
                                                               imag
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;</pre>
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
```

```
class Complex /* File: complex.h */
                                           10
                                                 10
 private:
    float real; float imag;
 public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
                                                                      10
                                                                      10
                                                                imag
      real += x.real; imag += x.imag;
      return (*this):
                                                    this
    Complex* add2(const Complex& x) // Return by value using pointer
                                                             Output
      real += x.real; imag += x.imag;
                                                             (4,5)
      return this:
                                                             (4,5)
                                                             (4,5)
    Complex& add3(const Complex& x) // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
    Complex* add2(const Complex& x) // Return by value using pointer
                                                             Output
      real += x.real; imag += x.imag;
                                                             (4,5)
      return this;
                                                             (4,5)
                                                             (4,5)
    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;
                                                           u
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                                                              (4,5)
                                                              (4,5)
                                               Cursor here <
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                           W
                                                                     10
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                real
           Χ
                imag
```

```
class Complex /* File: complex.h */
                                                  5
 private:
    float real; float imag;
 public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                           W
                                                               imag
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                       10
                                                            w
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                                                imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                            (4,5)
    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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
                                                               imag
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
   Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this;
                                           (4,5)
   Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
                                                                      10
                                                           W
                                                               imag 10
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this;
                                           (4,5)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
    Complex* add2(const Complex& x)
                                                                      14
    // Return by value using pointer
                                           Output
                                                                      15
                                                                imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
class Complex /* File: complex.h */
                                                           u
 private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                               imag
                                                                     10
      real += x.real; imag += x.imag;
      return (*this);
                                                    this
    Complex* add2(const Complex& x)
                                                                      14
    // Return by value using pointer
                                           Output
                                                                     15
                                                               imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
                                                       temp
    Complex& add3(const Complex& x)
                                                                     15
                                                               imag
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                            W
                                                                      10
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(\&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w)) \rightarrow print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                                                                      14
                                                        temp
           Х
                                                                      15
```

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                           w
                                                                      10
                                                               imag
      real += x.real; imag += x.imag;
      return (*this):
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                                               imag
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
                                                    this
      return this:
                                           (4,5)
                                                        temp
    Complex& add3(const Complex& x)
                                           (14,15)
                                                               imag
                                                                     15
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                                      10
                                                                      10
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(\&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w)) \rightarrow print(); // Complex* temp = this = &v
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
           Х
                                                        temp
                      15
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &v
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                      15
```

```
class Complex /* File: complex.h */
                                                  5
  private:
   float real; float imag;
  public:
   Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           w
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
   Complex* add2(const Complex& x)
                                                                      14
                                                           Х
    // Return by value using pointer
                                           Output
                                                               imag
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                                    this
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
                                                               imag
      return (*this);
    Complex* add2(const Complex& x)
                                                           Х
    // Return by value using pointer
                                           Output
                                                               imag
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
                                                    this
      return this:
                                           (4,5)
    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"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                                      10
                                                                      10
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
           Х
                      15
                                 imag
```

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                                               imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
                                                    this
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                            w
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                      14
                                                                real
                                                           Х
    // Return by value using pointer
                                           Output
                                                                imag
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                                    this
                                           (4,5)
      return this:
                                           (4,5)
                                                                real
                                                                      14
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                      5
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                     10
                                                           w
                                                                     10
                                                               imag
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                           Х
    // Return by value using pointer
                                           Output
                                                               imag
                                                                     15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
                                                    this
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                     15
                                                               imag
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                      14
                                                           Χ
    // Return by value using pointer
                                           Output
                                                                     15
                                                               imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                                    this
                                           (4,5)
      return this:
                                           (4,5)
                                                                      14
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                      15
                                                               imag
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                            u
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                        This is &y, i.e. address of y
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(\&u'); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl: // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0;
                imag
                                  imag
```

```
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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
   Complex* add2(const Complex& x)
                                                                      14
                                                                real
                                                           Х
    // Return by value using pointer
                                           Output
                                                                imag
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
                                                    this
      return this:
                                           (4,5)
                                                                real
                                                                      14
   Complex& add3(const Complex& x)
                                           (14,15)
                                                                      15
                                                                imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                           W
                                                                      10
                                                               imag
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
                                                  real
                      15
                                 imag
                                       15
```

```
class Complex /* File: complex.h */
                                                  5
                                                           u
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                      14
    // Return by value using pointer
                                           Output
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this;
                                           (4,5)
                                                                      14
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                      15
                                                               imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                real
      return (*this):
};
```

```
class Complex /* File: complex.h */
                                                           u
 private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                                                     15
                                                               imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
                                                                      14
    Complex& add3(const Complex& x)
                                           (14,15)
                                                               imag 15
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                           W
                                                                     10
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                                       14
                                                  real
           Х
                      15
                                 imag
                                       15
                                                         5
```

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           w
                                                                      10
                                                               imag
      real += x.real; imag += x.imag;
      return (*this):
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                                               imag
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
                                                               imag
                                                                      15
    // Return by reference
                                           (14,15)
                                                     this
      real += x.real: imag += x.imag:
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                      14
    // Return by value using pointer
                                           Output
                                                                      15
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
                                                                      14
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                     15
                                                               imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                      14
                                                                real
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                                                      15
                                                               imag
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
                                                                      14
    Complex& add3(const Complex& x)
                                           (14,15)
                                                               imag
                                                                     15
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                real
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                            Χ
                                                                       10
                                                            w
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                       14
                                                                 real
                                                            Х
    // Return by value using pointer
                                            Output
                                                                       15
                                            (4,5)
      real += x.real; imag += x.imag;
                                            (4,5)
      return this:
                                            (4,5)
                                                                       14
    Complex& add3(const Complex& x)
                                            (14,15)
                                                                       15
    // Return by reference
                                            (14,15)
                                                     this
      real += x.real; imag += x.imag;
                                                                       14
                                                                 real
      return (*this);
                                                                       15
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
void f(const Complex a) { a.print(); } // const Complex a = u
void g(const Complex* a) { a->print(); } // const Complex* a = &u
void h(const Complex& a) { a.print(); } // const Complex& a = u
int main()
                                                           W
                                                                     10
 // Check the paramete This is z ing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add(1(w)).print()); // Complex temp = *this = x
 Complex y(4, 5); (y.add(2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
                      14
                                                        14
                                                  real
                      15
                                       15
                                                        15
```

```
class Complex /* File: complex.h */
  private:
                                                                imag
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                       10
                                                            W
                                                                       10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                       14
                                                                 real
    // Return by value using pointer
                                            Output
                                                                      15
                                            (4,5)
      real += x.real; imag += x.imag;
                                            (4,5)
      return this:
                                            (4,5)
                                                                       14
    Complex& add3(const Complex& x)
                                                     this
                                            (14,15)
                                                                      15
                                                                imag
    // Return by reference
                                            (14,15)
                                            (14, 15)
      real += x.real; imag += x.imag;
                                                                       14
                                                                 real
      return (*this):
                                                                imag
                                                                       15
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                           u
#include "complex.h"
                                              Output
                                              (4,5)
void f(const Complex a) { a.print(); }
                                              (4,5)
                                              (4,5)
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                 Cursor here
                                                                      10
                                              (14,15)
                                                                real
                                                                      10
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                                                         14
                                                   real
                      15
                                        15
                                                         15
```

```
u x y z
ects u, x, y
```

```
#include <iostream> /* File: complex-test.cpp */
                                                            As the objects u. x. v
using namespace std;
                                                            and z are not used in
#include "complex.h"
                                                            the remaining code,
void f(const Complex a) { a.print(); }
                                                            their corresponding
void g(const Complex* a) { a->print(); }
                                                            boxes are not drawn
void h(const Complex& a) { a.print(); }
                                                            from now.
int main()
                                                                       10
                                                            W
                                                                       10
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
```

return 0:

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
                                              Output
                                              (4,5)
void f(const Complex a) { a.print(); }
                                              (4,5)
                                              (4,5)
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                 Cursor here
                                              (14,15)
                                                           w
                                                                      10
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
            a
                 imag
```

```
u x y z
class Complex /* File: complex.h */
                                                  5
                                            4
  private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
                                                     this
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                imag
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
    // Return by reference
                                           (14,15)
                                           (14, 15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
                                                     this
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
    // Return by reference
                                           (14, 15)
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
                                              Output
                                              (4,5)
void f(const Complex a) { a.print(); }
                                              (4,5)
                                             (4,5)
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                             (14,15)
int main()
                                                                      10
                                             (14,15)
                                                           w
                                                                     10
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
            а
```

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                               imag
                                                                      10
      real += x.real; imag += x.imag;
                                                     this
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
u x y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
                                                     this
      return (*this);
                                                                      14
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                               imag
                                                                      10
      real += x.real; imag += x.imag;
                                                     this
      return (*this);
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
    Complex& add3(const Complex& x)
                                           (14,15)
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                     10
      real += x.real; imag += x.imag;
                                                     this
      return (*this);
                                                                     14
                                                                real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                     15
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                           (4,5)
      return this:
                                           (4,5)
                                                                      14
                                                       temp1
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                     15
                                                               imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this);
};
                                                                     u x y z
class Complex /* File: complex.h */
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
                                              Output
                                              (4,5)
void f(const Complex a) { a.print(); }
                                              (4,5)
                                              (4,5)
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                              (14,15)
                                                            W
                                                                       10
                                                                imag
 // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
 Complex w(10, 10); c(\frac{This is temp1}{} dl << end1;
  Complex x(4, 5); (x.add1/(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add 2(w)) - print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.ad d(3(w)).print()); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                                                  14
                                           real
            а
                                   temp1
                       15
                                                 15
```

```
private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                                                      14
    Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                                                      15
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                                     this
                                           (4,5)
      return this:
                                           (4,5)
                                                                      14
                                                        temp1
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                      15
                                                                imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u \times y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                            w
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                                     this
                                           (4,5)
      return this:
                                           (4,5)
                                                        temp1
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                      15
                                                                imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           w
                                                                      10
      real += x.real; imag += x.imag;
      return (*this):
                                                                      14
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                               imag
                                                                     15
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                                     this
                                           (4,5)
      return this:
                                           (4,5)
                                                                      24
                                                                real
                                                       temp1
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                      25
                                                               imag
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
                                                                      10
                                                           w
                                                                      10
                                                                imag
      real += x.real; imag += x.imag;
      return (*this);
                                                                      14
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                                imag
                                                                      15
                                           Output
                                           (4,5)
      real += x.real; imag += x.imag;
                                                     this
                                           (4,5)
      return this:
                                           (4,5)
                                                                      24
                                                        temp1
    Complex& add3(const Complex& x)
                                           (14,15)
                                                                imag
                                                                      25
    // Return by reference
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                real
                                                                      24
      return (*this);
                                                        temp2
                                                                      25
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
using namespace std;
#include "complex.h"
                                              Output
                                              (4,5)
void f(const Complex a) { a.print(); }
                                             (4,5)
                                              (4,5)
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                             (14,15)
int main()
                                                                      10
                                              (14,15)
                                                                      10
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cc This is temp2 dl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl: // What is the output now?
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                                                                     24
            а
                                   temp1
                                                       temp2
                                                                     25
```

```
u \times y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                       10
                                                                 real
                                                            w
                                           Output
                                                                       10
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this);
                                           (4,5)
                                           (4,5)
    Complex* add2(const Complex& x)
                                                            a
    // Return by value using pointer
                                           (14,15)
                                           (14, 15)
                                           (14,15)
      real += x.real; imag += x.imag;
      return this:
                                           (24,25)
                                                                 real
                                                        temp1
    Complex& add3(const Complex& x)
                                                                       25
                                                                imag
    // Return by reference
                                                     this
      real += x.real; imag += x.imag;
      return (*this):
                                                        temp2
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
                                                                                            #include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                                                            using namespace std;
#include "complex.h"
                                                                                            #include "complex.h"
                                              Output
                                                                                                                                          Output
                                              (4,5)
                                                                                                                                          (4,5)
                                                                                            void f(const Complex a) { a.print(); }
void f(const Complex a) { a.print(); }
                                              (4,5)
                                                                                                                                          (4,5)
                                                                                                                                          (4,5)
                                              (4,5)
                                                                                            void g(const Complex* a) { a->print(); }
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                                                                            void h(const Complex& a) { a.print(); }
                                              (14,15)
                                                                                                                                          (14,15)
                                              (14,15)
                                                                                                                                          (14,15)
int main()
                                                                                            int main()
                                                                      10
                                                                                                                                                                  10
                                              (14,15)
                                                                                                                                          (14,15)
                                                           w
                                                                                                                                                       w
                                                                      10
                                                                                                                                                                  10
                                                                                                                                                            imag
  // Check the parameter passing methods
                                                                                             // Check the parameter passing methods
                                              (24,25)
                                                                                                                                          (24,25)
  Complex u(4, 5); f(u); g(&u); h(u);
                                                                                              Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
                                                                                              // Check the parameter returning methods
  Complex w(10, This is temp1 | t << This is temp2 endl;
                                                                                             Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)). rint(); // Complex temp = *this = x
                                                                                              Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y \cdot add2(w)) / > print(); // Complex* temp = this = &y
                                                                                              Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
                                                                                              Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?
                                                                                              cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print();
                                                 a.print(); cout << endl;</pre>
                                                                                              Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
                                                                                              Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
                                                                                             Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                                                                                              return 0:
                       14
                                                                                                                   14
                                                  24
                                                                                                             real
            а
                                                        temp2
                                                                                                        а
                                   temp1
                       15
                                                 25
                                                                                                                   15
                                                                      25
```

```
u x y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this);
                                           (4,5)
                                                     this
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
      return this:
                                           (24,25)
                                           (14,15)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                              Output
#include "complex.h"
                                              (4,5)
                                              (4,5)
                                              (4,5)
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
                                              (14,15)
int main()
                                 Cursor here
                                                                      10
                                                                real
                                              (24,25)
                                                                      10
 // Check the parameter passing methods
                                              (14,15)
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                       14
            а
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                              Output
#include "complex.h"
                                              (4,5)
                                              (4,5)
                                             (4,5)
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
                                              (14,15)
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                                                      10
                                                           W
                                              (24,25)
                                                                      10
 // Check the parameter passing methods
                                              (14,15)
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
 cout << endl << endl; // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
                       14
                                  real
                             h
            а
                       15
```

```
class Complex /* File: complex.h */
                                                  5
 private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                                                imag
                                           (4,5)
      return (*this):
                                           (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                                imag
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
u x y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this);
                                            (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
      return this:
                                                                       4
                                                                real
                                                           b
                                           (24,25)
                                           (14,15)
                                                                       5
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                              Output
#include "complex.h"
                                              (4,5)
                                              (4,5)
                                              (4,5)
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
                                              (14,15)
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                                                      10
                                                                real
                                              (24,25)
                                                                      10
 // Check the parameter passing methods
                                              (14,15)
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                              b
            а
                       15
```

```
private:
   float real; float imag;
  public:
   Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
                                                                      14
   Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                               imag
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
                                                                      5
   Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
                                                                      u x y z
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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                           Х
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
                                                                      14
   Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                                                      15
                                           (14,15)
```

(14,15)

(14,15)

(24,25)

(14,15)

this

b

14

15

class Complex /* File: complex.h */

real += x.real; imag += x.imag;

real += x.real; imag += x.imag;

Complex& add3(const Complex& x)

return this:

// Return by reference

return (*this):

};

u x y z

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
                                                                      10
                                                           W
                                           Output
                                                                      10
                                                               imag
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                               imag
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      14
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
```

 $u \times v z$

```
u \times y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                            W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      14
      return this:
                                                                real
                                                            b
                                           (24,25)
                                           (14,15)
                                                                      15
                                                                imag
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                              Output
#include "complex.h"
                                              (4,5)
                                              (4,5)
                                              (4,5)
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
                                              (14,15)
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                                                       10
                                                            W
                                              (24,25)
                                                                      10
  // Check the parameter passing methods
                                              (14,15)
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); C( This is &b, i.e. address of b ndl;
  Complex x(4, 5); (x.add1(w)), print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                                         14
                       14
                              h
            а
                                  imag
                       15
                                         15
                                                                      u x y z
```

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
                                                                      10
                                                           W
                                           Output
                                                                      10
                                                               imag
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this):
                                           (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                               imag
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      14
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
                                                                      15
                                                               imag
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                            Х
                                                                      10
                                                            W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                            (4,5)
                                            (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                            a
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      24
      return this:
                                                                real
                                                            b
                                           (24,25)
                                           (14,15)
                                                                      15
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u \times y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                            W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
    Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
      return this:
                                                                      24
                                                            b
                                           (24,25)
                                           (14,15)
                                                                      25
                                                                imag
    Complex& add3(const Complex& x)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
class Complex /* File: complex.h */
                                                                                           using namespace std;
                                                                                           #include "complex.h"
  private:
    float real; float imag;
  public:
   Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                                           int main()
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
                                                                      14
                                                                real
   Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                               imag
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      24
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
    // Return by reference
                                                                                             return 0:
      real += x.real; imag += x.imag;
                                                                                                                   14
                                                                                                        а
      return (*this);
                                                                                                                   15
};
                                                                      u x y z
class Complex /* File: complex.h */
                                                                                           using namespace std;
  private:
                                                                                           #include "complex.h"
    float real; float imag;
```

```
#include <iostream> /* File: complex-test.cpp */
                                              Output
                                              (4,5)
                                              (4,5)
                                              (4,5)
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
                                              (14,15)
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
                                                                       10
                                                            W
                                              (24,25)
                                                                       10
 // Check the parameter passing methods
                                              (14,15)
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
 Complex w(10, 10); C( This is &b, i.e. address of b !ndl;
  Complex x(4, 5); (x.add1(w)), print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w)) - print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
                                         24
                                   real
                              h
                                  imag
                                         25
```

```
public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                            W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this);
                                            (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                            a
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      24
      return this:
                                                                real
                                                            b
                                           (24,25)
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
#include <iostream> /* File: complex-test.cpp */
                                              Output
                                              (4,5)
                                              (4,5)
                                              (4,5)
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
                                              (14,15)
void h(const Complex& a) { a.print(); }
                                              (14,15)
                                              (14,15)
int main()
                                                                      10
                                                                real
                                                           W
                                              (24,25)
                                                                imag
                                                                      10
 // Check the parameter passing methods
                                              (14,15)
 Complex u(4, 5); f(u); g(&u); h(u);
                                              (24,25)
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
                       14
                                         24
                              b
            а
                       15
```

```
u x y z
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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
                                                                      14
   Complex* add2(const Complex& x)
                                                           а
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                     this
                                                                      24
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                           (24,25)
      real += x.real; imag += x.imag;
      return (*this);
};
                                                                      u x y z
                                                 Output
```

```
u x y z
                                                 Output
#include <iostream> /* File: complex-test.cpp */ ( 4 , 5 )
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14, 15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                     10
                                                               real
                                                          W
                                                 (24,25)
                                                                     10
                                                 (24,25)
                                                               imag
 // Check the parameter passing methods
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                                        24
                       14
                                  real
                             h
            а
                       15
                                  imag
                                        25
```

```
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14,15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                      10
                                                           W
                                                 (24,25)
                                                                      10
 // Check the parameter passing methods
                                                 (24,25)
 Complex u(4, 5); f(u); g(&u); h(u);
 // Check the parameter returning methods
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print();
                                                c.print():
 return 0;
                       14
                                                   real
                              b
            а
                                               С
                       15
                                         25
                 imag
```

```
u x y z
class Complex /* File: complex.h */
                                           4
                                                  5
 private:
    float real; float imag;
  public:
    Complex(float r, float i) { real = r; imag = i; }
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this);
                                           (4,5)
                                           (4,5)
    Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
      return this:
                                                                      24
                                                           b
                                           (24,25)
                                           (14,15)
                                                                      25
                                                               imag
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                           (24,25)
                                                    this
      real += x.real; imag += x.imag;
      return (*this):
};
```

```
u x y z
                                                                                                                                             Output
                                                                                           #include <iostream> /* File: complex-test.cpp */ ( 4 , 5 )
class Complex /* File: complex.h */
                                                                                           using namespace std;
                                                                                                                                             (4,5)
                                                                                                                                             (4,5)
                                                                                           #include "complex.h"
  private:
   float real; float imag;
                                                                                                                                             ( 14 , 15 )
  public:
                                                                                           void f(const Complex a) { a.print(); }
                                                                                                                                             (14,15)
   Complex(float r, float i) { real = r; imag = i; }
                                                                                           void g(const Complex* a) { a->print(); }
                                                                                                                                             (14,15)
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
                                                                                           void h(const Complex& a) { a.print(); }
                                                                                                                                             (24,25)
                                                                                                                                             (14,15)
   Complex add1(const Complex& x) // Return by value
                                                                                           int main()
                                                                      10
                                                           W
                                                                                                                                                      W
                                                                                                                                             (24,25)
                                           Output
                                                                      10
                                                                                                                                                           imag
      real += x.real; imag += x.imag;
                                                                                             // Check the parameter passing methods
                                                                                                                                             (24,25)
      return (*this);
                                           (4,5)
                                                                                             Complex u(4, 5); f(u); g(&u); h(u);
                                           (4,5)
                                                                                             // Check the parameter returning methods
                                           (4,5)
                                                                      14
   Complex* add2(const Complex& x)
                                                                                             Complex w(10, 10); cout << endl << endl;
    // Return by value using pointer
                                                                                             Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
                                           (14,15)
                                                                      15
                                           (14,15)
                                                                                             Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                                             Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
                                                                      24
      return this:
                                                                                             cout << endl << endl; // What is the output now?
                                                                real
                                                           b
                                           (24,25)
                                                                                             Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
                                                                                             Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
                                           (24,25)
    // Return by reference
                                                                                             Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
                                           (24,25)
                                                    this
                                                                                             return 0:
                                                                real
      real += x.real; imag += x.imag;
                                                                                                                   14
                                                                                                                                    24
                                                                                                                                               real
                                                                      5
                                                                                                        а
                                                                                                                         h
                                                                                                                                          С
      return (*this);
                                                                                                                  15
                                                                                                                                    25
                                                                                                                                                     5
};
                                                                      u x y z
class Complex /* File: complex.h */
                                                                                           class Complex /* File: complex.h */
  private:
                                                                                             private:
                                                                                               float real; float imag;
   float real; float imag;
  public:
                                                                                             public:
   Complex(float r, float i) { real = r; imag = i; }
                                                                                               Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
```

10

10

14

15

24

25

5

W

b

Output

(4,5)

(4,5)

(4,5)

(14,15)

(14,15)

(14,15)

(24,25)

(14,15)

(24,25)

(24, 25) this

Complex add1(const Complex& x) // Return by value

real += x.real; imag += x.imag;

Complex* add2(const Complex& x)

// Return by value using pointer

Complex& add3(const Complex& x)

real += x.real; imag += x.imag;

real += x.real; imag += x.imag;

return (*this);

return this:

// Return by reference

return (*this):

};

```
void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                           w
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this);
                                           (4,5)
                                           (4,5)
                                           (4,5)
    Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                      24
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
                                                                imag
                                                                      25
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                           (24,25)
                                                     this
                                                                      14
      real += x.real; imag += x.imag;
      return (*this):
};
```

10

10

 $u \times y z$

```
u x y z
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; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                      10
                                                           W
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
      return (*this):
                                           (4,5)
                                           (4,5)
                                           (4,5)
                                                                      14
   Complex* add2(const Complex& x)
    // Return by value using pointer
                                           (14,15)
                                                                      15
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                      24
      return this:
                                           (24,25)
                                                           b
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                           (24,25)
                                                     this
                                                                      14
                                                                real
      real += x.real: imag += x.imag:
                                                                      15
      return (*this);
};
                                                                      u x y z
```

```
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                real
                                                                      10
                                                           w
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                                               imag
                                           (4,5)
      return (*this):
                                           (4,5)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                           (14,15)
                                                               imag
                                                                      15
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                      24
      return this:
                                                           b
                                           (24,25)
                                           (14,15)
                                                               imag
                                                                      25
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                           (24,25)
                                                    this
                                                                real
                                                                      14
      real += x.real; imag += x.imag;
                                                                      15
      return (*this);
                                                                imag
};
```

 $u \times v z$

```
Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                  (4,5)
                                                   (4,5)
#include "complex.h"
                                                  (14,15)
void f(const Complex a) { a.print(); }
                                                  (14,15)
void g(const Complex* a) { a->print(); }
                                                   (14,15)
void h(const Complex& a) { a.print(); }
                                                  (24,25)
                                                  (14,15)
int main()
                                                                        10
                                                             W
                                                   (24,25)
                                                                        10
  // Check the parameter passing methods
                                                  (24,25)
  Complex u(4, 5); f(u); g(&u); h(u);
  // Check the parameter returning methods
  Complex w(10, 10); cout << endl << endl;</pre>
  Complex x(4, 5); (x. This isc )).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add^2(w)) \rightarrow print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.ad d(3(w)).print()); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add^{\dagger}_{2}(w)->add2(w)->print(); b.print(); cout << endl;
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0;
                        14
                                                     real
                              b
                                                С
             а
                        15
                                          25
                 imag
```

```
u x y z
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; }</pre>
    Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                real
                                                           w
                                           Output
                                                                      10
      real += x.real; imag += x.imag;
                                           (4,5)
      return (*this);
                                           (4,5)
                                           (4,5)
    Complex* add2(const Complex& x)
                                                           a
    // Return by value using pointer
                                                                      15
                                           (14,15)
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
      return this:
                                                                      24
                                                           b
                                           (24,25)
                                           (14,15)
                                                                      25
                                                               imag
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                           (24,25)
                                                    this
                                                                      14
      real += x.real; imag += x.imag;
                                                                      15
      return (*this):
                                                               imag
};
```

```
class Complex /* File: complex.h */
                                                                                           class Complex /* File: complex.h */
  private:
                                                                                             private:
    float real; float imag;
                                                                                               float real; float imag;
  public:
                                                                                             public:
   Complex(float r, float i) { real = r; imag = i; }
                                                                                               Complex(float r, float i) { real = r; imag = i; }
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
                                                                                               void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
   Complex add1(const Complex& x) // Return by value
                                                                                               Complex add1(const Complex& x) // Return by value
                                                                      10
                                                                                                                                                           real
                                                                                                                                                                 10
                                                           W
                                                                                                                                                      W
                                           Output
                                                                     10
                                                                                                                                      Output
                                                                                                                                                                 10
                                                                                                                                                           imag
      real += x.real; imag += x.imag;
                                                                                                 real += x.real; imag += x.imag;
                                                                                                                                      (4,5)
      return (*this);
                                           (4,5)
                                                                                                 return (*this):
                                           (4,5)
                                                                                                                                      (4,5)
                                           (4,5)
                                                                                                                                      (4,5)
                                                                     14
                                                                                                                                                                 14
                                                                real
    Complex* add2(const Complex& x)
                                                                                               Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                                               // Return by value using pointer
                                           (14,15)
                                                                     15
                                                                                                                                                           imag
                                                                                                                                                                 15
                                                                                                                                      (14,15)
                                           (14,15)
                                                                                                                                      (14,15)
                                           (14,15)
                                                                                                                                      (14,15)
      real += x.real; imag += x.imag;
                                                                                                 real += x.real; imag += x.imag;
                                                                      24
                                                                                                                                                                 24
      return this:
                                                                                                 return this:
                                           (24,25)
                                                           b
                                                                                                                                                      b
                                                                                                                                      (24,25)
                                           (14,15)
                                                                      25
                                                                                                                                      (14,15)
                                                                                                                                                           imag
                                                                                                                                                                 25
    Complex& add3(const Complex& x)
                                                                                               Complex& add3(const Complex& x)
                                           (24,25)
                                                                                                                                      (24,25)
    // Return by reference
                                                                                               // Return by reference
                                           (24,25)
                                                    this
                                                                                                                                      (24,25)
                                                                                                                                                this
                                                                      24
                                                                                                                                                           real
                                                                                                                                                                 24
      real += x.real; imag += x.imag;
                                                                                                 real += x.real; imag += x.imag;
                                                                      15
                                                                                                                                                                 25
                                                                                                                                                           imag
      return (*this);
                                                                                                 return (*this);
};
                                                                                           };
                                                                     u x y z
                                                                                                                                                                 u x y z
                                                                                           #include <iostream> /* File: complex-test.cpp */(4.5)
class Complex /* File: complex.h */
                                                                                           using namespace std;
                                                                                                                                            (4,5)
                                                                                                                                             (4,5)
  private:
                                                                                           #include "complex.h"
    float real; float imag;
                                                                                                                                             (14,15)
  public:
                                                                                           void f(const Complex a) { a.print(); }
                                                                                                                                            (14,15)
   Complex(float r, float i) { real = r; imag = i; }
                                                                                           void g(const Complex* a) { a->print(); }
                                                                                                                                             (14,15)
   void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
                                                                                           void h(const Complex& a) { a.print(); }
                                                                                                                                             (24,25)
                                                                                                                                            (14,15)
    Complex add1(const Complex& x) // Return by value
                                                                                           int main()
                                                                      10
                                                                                                                                                                 10
                                                                                                                                                           real
                                                           W
                                                                                                                                                      W
                                                                                                                                             (24,25)
                                           Output
                                                                      10
                                                                                                                                                                 10
                                                                                                                                            (24,25)
      real += x.real; imag += x.imag;
                                                                                             // Check the parameter passing methods
                                           (4,5)
      return (*this);
                                                                                             Complex u(4, 5); f(u); g(&u); h(u);
                                           (4,5)
                                                                                             // Check the parameter returning methods
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                                                             Complex w(10, 10); cout << endl << endl;</pre>
                                                           а
    // Return by value using pointer
                                                                                             Complex x(4, 5); (x. This isc)).print(); // Complex temp = *this = x
                                                                      15
                                           (14,15)
                                                                                             Complex y(4, 5); (y.add(w))-print(); // Complex* temp = this = &y
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                                             Complex z(4, 5); (z.add^3(w)).print(); // Complex& temp = *this = z
      return this:
                                                                     24
                                                                                             cout << endl << endl: // What is the output now?
                                                                real
                                                           b
                                           (24,25)
                                                                                             Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
                                                                                             Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
                                           (24,25)
    // Return by reference
                                                                                             Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
                                           (24,25)
                                                    this
                                                                                             return 0;
                                                                      24
      real += x.real; imag += x.imag;
                                                                                                                  14
                                                                                                                                    24
                                                                                                                                                     24
                                                                                                                         b
                                                                                                                                          С
      return (*this):
                                                                      25
                                                                                                        а
                                                                                                                  15
                                                                                                                                    25
```

};

u x y z

 $u \times v z$

```
u x y z
                                                                                                                                                                 u x y z
                                                                                                                                             Output
                                                                                           #include <iostream> /* File: complex-test.cpp */ ( 4 , 5 )
class Complex /* File: complex.h */
                                                                                           using namespace std;
                                                                                                                                             (4,5)
                                                                                                                                             (4,5)
                                                                                           #include "complex.h"
  private:
    float real; float imag;
                                                                                                                                             (14,15)
                                                                                           void f(const Complex a) { a.print(); }
  public:
                                                                                                                                             (14, 15)
    Complex(float r, float i) { real = r; imag = i; }
                                                                                           void g(const Complex* a) { a->print(); }
                                                                                                                                             (14,15)
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
                                                                                           void h(const Complex& a) { a.print(); }
                                                                                                                                             (24,25)
                                                                                                                                             (14,15)
    Complex add1(const Complex& x) // Return by value
                                                                                           int main()
                                                                                                                                                                 10
                                                                      10
                                                                                                                                                           real
                                                           W
                                                                                                                                                      W
                                                                                                                                             (24,25)
                                           Output
                                                                      10
                                                                                                                                                                 10
                                                                                                                                                           imag
      real += x.real; imag += x.imag;
                                                                                             // Check the parameter passing methods
                                                                                                                                             (24,25)
      return (*this);
                                           (4,5)
                                                                                             Complex u(4, 5); f(u); g(&u); h(u);
                                                                                                                                             (24,25)
                                           (4,5)
                                                                                             // Check the parameter returning methods
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                                                             Complex w(10, 10); cout << endl << endl;
                                                           а
    // Return by value using pointer
                                                                                             Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
                                           (14,15)
                                                                      15
                                           (14,15)
                                                                                             Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                                             Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
                                                                      24
      return this:
                                                                                             cout << endl << endl; // What is the output now?
                                                           b
                                           (24,25)
                                                                                             Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
                                           (14,15)
                                                                      25
                                                                                             Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
    Complex& add3(const Complex& x)
                                           (24,25)
    // Return by reference
                                                                                             Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
                                           (24,25)
                                                     this
                                                                                             return 0:
                                                                      24
                                           (24,25)
      real += x.real; imag += x.imag;
                                                                                                                   14
                                                                                                                                    24
                                                                                                                                                     24
                                                                      25
                                                                                                                         h
                                                                                                        а
                                                                                                                                          С
      return (*this);
                                                                                                                   15
                                                                                                                                    25
                                                                                                                                                     25
};
                                                                      u x y z
                                                                                                                                                                 u x y z
                                                                                           #include <iostream> /* File: complex-test.cpp */(4.5)
class Complex /* File: complex.h */
                                                                                           using namespace std;
                                                                                                                                             (4,5)
                                                                                                                                             (4,5)
  private:
                                                                                           #include "complex.h"
    float real; float imag;
                                                                                                                                             (14,15)
  public:
                                                                                           void f(const Complex a) { a.print(); }
                                                                                                                                             (14,15)
    Complex(float r, float i) { real = r; imag = i; }
                                                                                           void g(const Complex* a) { a->print(); }
                                                                                                                                             (14,15)
    void print() { cout << "( " << real << " , " << imag << " )" << endl; }</pre>
                                                                                           void h(const Complex& a) { a.print(); }
                                                                                                                                             (24,25)
                                                                                                                                             (14,15)
    Complex add1(const Complex& x) // Return by value
                                                                                           int main()
                                                                      10
                                                                                                                                                                 10
                                                                                                                                                           real
                                                           W
                                                                                                                                                      W
                                                                                                                                             (24,25)
                                           Output
                                                                      10
                                                                                                                                                           imag
                                                                                                                                                                 10
                                                                                                                                             (24,25)
      real += x.real; imag += x.imag;
                                                                                             // Check the parameter passing methods
                                           (4,5)
      return (*this);
                                                                                             Complex u(4, 5); f(u); g(&u); h(u);
                                                                                                                                             (24,25)
                                           (4,5)
                                                                                             // Check the parameter returning methods
                                                                                                                                             (24,25)
                                           (4,5)
                                                                      14
    Complex* add2(const Complex& x)
                                                                                             Complex w(10, 10); cout << endl << endl;</pre>
                                                           а
    // Return by value using pointer
                                                                                             Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
                                                                      15
                                           (14,15)
                                                                                             Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
                                           (14,15)
                                           (14,15)
      real += x.real; imag += x.imag;
                                                                                             Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
      return this;
                                                                      24
                                                                                             cout << endl << endl: // What is the output now?</pre>
                                                           b
                                           (24,25)
                                                                                             Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
                                           (14,15)
                                                                      25
    Complex& add3(const Complex& x)
                                                                                             Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
                                           (24,25)
    // Return by reference
                                                                                             Complex c(4, 5); c.add3(w).add3(w).print();
                                                                                                                                            c.print():
                                           (24,25)
                                                     this
                                                                                             return 0;
                                                                      24
                                                                real
      real += x.real; imag += x.imag;
                                           (24,25)
                                                                                                                   14
                                           (24,25)
                                                                                                                         b
                                                                      25
                                                                                                                                          С
      return (*this):
                                                                                                        а
                                                                                                                   15
                                                                                                                                    25
};
```

```
u x y z
                                                 Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14, 15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                      10
                                                           W
                                                 (24,25)
                                                                     10
  // Check the parameter passing methods
                                                 (24,25)
  Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
  // Check the parameter returning methods
                                                 (24,25)
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print();
                                                a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
                       14
            а
                       15
                                        25
```

```
u \times v z
                                                 Output
#include <iostream> /* File: complex-test.cpp */ ( 4 , 5 )
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14, 15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                      10
                                                                real
                                                           W
                                                 (24,25)
                                                                      10
                                                               imag
 // Check the parameter passing methods
                                                 (24,25)
  Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
  // Check the parameter returning methods
                                                 (24,25)
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
```

```
u x y 🗶
                                                 Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14,15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                     10
                                                           W
                                                 (24,25)
                                                                     10
                                                 (24,25)
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
 // Check the parameter returning methods
                                                 (24,25)
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
u x 💥
                                                 Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h'
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14,15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                     10
                                                               real
                                                          W
                                                 (24,25)
                                                                     10
                                                 (24,25)
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
 // Check the parameter returning methods
                                                 (24,25)
 Complex w(10, 10); cout << endl << endl;
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
  cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
L.... U
```

```
Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14, 15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                                      10
                                                           W
                                                 (24,25)
                                                                     10
  // Check the parameter passing methods
                                                 (24,25)
  Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
  // Check the parameter returning methods
                                                 (24,25)
  Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0:
```

```
Output
#include <iostream> /* File: complex-test.cpp */ ( 4 , 5 )
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14, 15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                 (24,25)
 // Check the parameter passing methods
                                                 (24,25)
  Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
  // Check the parameter returning methods
                                                 (24,25)
 Complex w(10, 10); cout << endl << endl;
  Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; // What is the output now?</pre>
  Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0:
```

```
Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14,15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                 (24,25)
 // Check the parameter passing methods
                                                 (24,25)
 Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
 // Check the parameter returning methods
                                                 (24,25)
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex  temp = *this = z
 cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```

```
Output
#include <iostream> /* File: complex-test.cpp */(4.5)
using namespace std;
                                                 (4,5)
                                                 (4,5)
#include "complex.h"
                                                 (14,15)
void f(const Complex a) { a.print(); }
                                                 (14,15)
void g(const Complex* a) { a->print(); }
                                                 (14,15)
void h(const Complex& a) { a.print(); }
                                                 (24,25)
                                                 (14,15)
int main()
                                                 (24,25)
                                                 (24,25)
 // Check the parameter passing methods
 Complex u(4, 5); f(u); g(&u); h(u);
                                                 (24,25)
 // Check the parameter returning methods
                                                 (24,25)
 Complex w(10, 10); cout << endl << endl;</pre>
 Complex x(4, 5); (x.add1(w)).print(); // Complex temp = *this = x
 Complex y(4, 5); (y.add2(w))-print(); // Complex* temp = this = &y
 Complex z(4, 5); (z.add3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl: // What is the output now?</pre>
 Complex a(4, 5); a.add1(w).add1(w).print(); a.print(); cout << endl;</pre>
 Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;</pre>
 Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
 return 0;
```