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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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
                                                   i
                                            r
                                                                 real
                                                                        4
                                                   5
                                            4
                                                            u
  private:
                                                                        5
                                                                 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
      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
                                            r
                                                                real
                                                                        4
                                                            u
  private:
                                                                 imag
                                                                        5
    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;
                                                                real
                                                           u
#include "complex.h"
                                                                imag
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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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()
                                                                real
  // Check the parameter passing methods
                                                                imag
  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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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()
                                                                real
                                                            a
  // Check the parameter passing methods
                                                                imag
  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 */
                                                                 real
                                                             u
  private:
                                                                        5
                                                                 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
                                                                 real
                                                                        4
                                                             a "
      real += x.real; imag += x.imag;
                                                                 imag
                                                                        5
      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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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()
                                                            a
  // Check the parameter passing methods
                                                                imag
  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;
                                                                real
                                                           u
#include "complex.h"
                                                                imag
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 */
                                                     a
                                                                real
using namespace std;
                                                           u
#include "complex.h"
                                                                imag
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 */
                                                     а
                                                                real
using namespace std;
                                                           u
#include "complex.h"
                                                                imag
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 */
                                                      a
                                                                real
                                                                        4
  private:
                                                                imag
                                                                        5
    float real; float imag;
  public:
                                                     this
    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 */
                                                      a
using namespace std;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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;
                                                                real
                                                           u
#include "complex.h"
                                                                imag
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;
                                                                real
                                                           u
#include "complex.h"
                                                                imag
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;
                                                                real
                                                           u
#include "complex.h"
                                                                imag
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 */
                                                                 real
                                                            U.
  private:
                                                                imag
                                                                        5
    float real; float imag;
  public:
                                                     this
    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;
      return this;
    Complex& add3(const Complex& x) // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
#include <iostream> /* File: complex-test.cpp */
                                                                real
using namespace std;
#include "complex.h"
                                                                imag
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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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()
                                                                real
                                                                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 */
                                            r
                                                                 real
                                                   10
                                            10
                                                            u
  private:
                                                                 imag
                                                                        5
    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
                                                                 imag
                                                                        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;
    Complex& add3(const Complex& x) // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
class Complex /* File: complex.h */
                                            r
                                                                 real
                                                   10
                                            10
                                                             u
  private:
                                                                        5
                                                                 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
                                                                 real
                                                                 imag
                                                                        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;
      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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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
                                                                real
                                                                imag
                                                                       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();
                                                               Output
  return 0;
                                                               (4,5)
                                                               (4,5)
                                                Cursor here
```

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

```
class Complex /* File: complex.h */
                                             r
                                                                 real
                                             4
                                                             u
  private:
                                                                 imag
                                                                         5
    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
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                      this
    Complex* add2(const Complex& x)
                                                                  real
    // Return by value using pointer
                                            Output
                                                                 imag
      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 */
                                                                 real
                                                             u
  private:
                                                                         5
                                                                 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
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                      this
    Complex* add2(const Complex& x)
                                                                 real
    // Return by value using pointer
                                            Output
                                                                 imag
      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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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
                                                                real
                                                                imag
                                                                       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;
                real
                imag
```

```
class Complex /* File: complex.h */
                                                                 real
                                                             u
  private:
                                                                 imag
                                                                         5
    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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                      this
    Complex* add2(const Complex& x)
                                                                 real
    // Return by value using pointer
                                            Output
                                                                 imag
      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 */
                                                                 real
                                                             u
  private:
                                                                 imag
                                                                         5
    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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                      this
    Complex* add2(const Complex& x)
                                                                        14
                                                                  real
    // Return by value using pointer
                                            Output
                                                                 imag
      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 */
                                                                 real
                                                             u
  private:
                                                                 imag
                                                                         5
    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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                      this
    Complex* add2(const Complex& x)
                                                                        14
                                                                  real
    // Return by value using pointer
                                            Output
                                                                 imag
                                                                        15
      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 */
                                                                       real
                                                                  u
  private:
                                                                      imag
                                                                              5
    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
                                                                  X
                                                                              10
                                                                       real
                                                                  W
                                                                      imag
                                                                              10
      real += x.real; imag += x.imag;
      return (*this);
                         Returning it by value, we need to construct a
                         temporary object "temp" which is a copy of x
                                                          this
    Complex* add2(const Complex& x)
                                                                              14
                                                                       real
    // Return by value using pointer
                                               Output
                                                                      imag
                                                                              15
      real += x.real; imag += x.imag;
                                               (4,5)
(4,5)
      return this;
                                                                              14
                                                                       real
                                                              temp
    Complex& add3(const Complex& x)
                                                                      imag
                                                                              15
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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
                                                                real
                         This is temp
                                                                imag
                                                                       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
                real
                                                         temp
                                                                imag
                                                                       15
                imag
```

```
class Complex /* File: complex.h */
                                                                 real
                                                             u
  private:
                                                                         5
                                                                 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
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                        14
                                                                 real
                                                             Χ
    // Return by value using pointer
                                            Output
                                                                 imag
                                                                        15
                                            (4,5)
      real += x.real; imag += x.imag;
                                            (4,5)
                                                      this
      return this;
                                            (4,5)
                                                                        14
                                                                 real
                                                          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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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()
                                                                real
                                                                       10
                         This is temp
                                                                imag
                                                                       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
                                                         temp
                imag
```

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

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

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

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

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

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

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

```
class Complex /* File: complex.h */
                                                                 real
                                                             u
  private:
                                                                         5
                                                                 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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                        14
                                                                 real
                                                             Χ
    // Return by value using pointer
                                            Output
                                                                 imag
                                                                        15
                                            (4,5)
      real += x.real; imag += x.imag;
                                            (4,5)
                                                      this
      return this;
                                            (4,5)
                                                                        14
                                                                 real
    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;
                                                                  real
                                                              u
#include "complex.h"
                                                                  imag
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
                                                                  real
                         This is &y, i.e. address of y
                                                                  imag
                                                                         10
  // Check the parameter passing methods
  Complex u(4, 5); f(u); g(\&\psi); 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.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
                 real
                                   real
                              У
            Χ
                                   imag
                 imag
```

```
class Complex /* File: complex.h */
                                                                  real
                                                             u
  private:
                                                                         5
                                                                 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
                                                                  real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                        14
                                                                  real
                                                             Χ
    // Return by value using pointer
                                            Output
                                                                 imag
                                                                        15
                                            (4,5)
      real += x.real; imag += x.imag;
                                            (4,5)
                                                      this
      return this;
                                                                        14
                                                                  real
    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;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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
                                                                 real
                                                                imag
                                                                       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;
                                                   real
                real
                                  real
                             У
                                               Z
                                  imag
                                                   imag
                imag
```

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

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

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                                real
                                                            u
#include "complex.h"
                                                                imag
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
                                                                 real
                                                                imag
                                                                       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;
                                                   real
                real
                                  real
                             У
                                               Ζ
                                  imag
                imag
                                                   imag
```

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

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

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

```
class Complex /* File: complex.h */
                                                                     real
                                                                u
  private:
                                                                             5
                                                                     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
                                                                X
                                                                            10
                                                                     real
                                                                W
                                                                     imag
                                                                            10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                            14
                                                                     real
                                                                Χ
    // Return by value using pointer
                                              Output
                                                                     imag
                                                                            15
      real += x.real; imag += x.imag;
                                              (4,5)
(4,5)
      return this;
                                                                            14
                                                                     real
    Complex& add3(const Complex& x)
                                              (14,15)
                                                                     imag
                                                                            15
    // Return by reference
                                              (14,15)
                                                         this
      real += x.real; imag += x.imag;
                                                                            14
                                                                     real
      return (*this);
                                                                     imag
                                                                            15
              Returning it by reference, i.e. returning z
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                                 real
                                                            u
#include "complex.h"
                                                                 imag
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
                                                                 real
                                                                 imag
                                                                        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.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
                                         14
                                                   real
                                                          14
                real
                             У
                                               Ζ
                                  imag
                                                          15
                imag
                                                   imag
```

```
class Complex /* File: complex.h */
                                                                  real
                                                             u
  private:
                                                                         5
                                                                  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
                                                                  real
                                                             W
                                                                  imag
                                                                         10
      real += x.real; imag += x.imag;
      return (*this);
    Complex* add2(const Complex& x)
                                                                         14
                                                                  real
                                                             Χ
    // Return by value using pointer
                                            Output
                                                                  imag
                                                                         15
                                            (4,5)
      real += x.real; imag += x.imag;
                                             (4,5)
      return this;
                                            (4,5)
                                                                         14
                                                                  real
                                                              У
    Complex& add3(const Complex& x)
                                                       this
                                            (14,15)
                                                                  imag
                                                                         15
    // Return by reference
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                         14
                                                                  real
      return (*this);
                                                              Ζ
                                                                  imag
                                                                         15
};
```

```
#include <iostream> /* File: complex-test.cpp */
using namespace std;
                                                                 real
                                                             u
#include "complex.h"
                                               Output
                                                                 imag
                                               (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)
                                  Cursor here
int main()
                                                                        10
                                                                 real
                                               (14,15)
                                                                 imag
                                                                        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;
                                         14
                                                           14
                real
                                  real
                                                    real
                              У
                                               Ζ
            Χ
                       15
                                  imag
                                                    imag
                                                           15
                imag
```

```
#include <iostream> /* File: complex-test.cpp */
                                                            As the objects u, x, y
using namespace std;
                                                             and z are not used in
#include "complex.h"
                                                            the remaining code,
                                                             their corresponding
void f(const Complex a) { a.print(); }
void g(const Complex* a) { a->print(); }
                                                             boxes are not drawn
void h(const Complex& a) { a.print(); }
                                                            from now.
int main()
                                                                 real
                                                                        10
                                                                 imag
                                                                        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;
```

```
#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)
                                  Cursor here
int main()
                                                                       10
                                                                real
                                              (14,15)
                                                                imag
                                                                       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
             a
                 imag
```

```
class Complex /* File: complex.h */
                                            r
                                                   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
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                                      this
      return (*this);
                                                                 real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                        5
                                                                 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);
};
```

```
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
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                                      this
      return (*this);
                                                                 real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                        5
                                                                 imag
                                            Output
      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);
```

**}**;

```
#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
                                                                real
                                              (14,15)
                                                                imag
                                                                       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
            a
                 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
                                                             X
                                                                        10
                                                                 real
                                                            W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                                      this
      return (*this);
                                                                 real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                 imag
                                                                        5
                                            Output
      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);
};
```

```
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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                                      this
      return (*this);
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                 imag
                                                                        5
                                            Output
      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);
};
```

```
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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                                      this
      return (*this);
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            Output
      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);
};
```

```
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
                                                                 X
                                                                             10
                                                                     real
                                                                 W
                                                                     imag
                                                                             10
      real += x.real; imag += x.imag;
                                                         this
      return (*this);
                        Returning it by value, we need to construct a
                        temporary object "temp1" which is a copy of a
                                                                             14
                                                                     real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                     imag
                                                                             15
                                               Output
                                               (4,5)
      real += x.real; imag += x.imag;
                                               (4,5)
      return this;
                                               (4,5)
                                                                             14
                                                                     real
                                                             temp1
    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"
                                               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
                                                                 real
                                               (14,15)
                                                                 imag
                                                                        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 temp1 dl << endl;
  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  (3(w)).print(); // Complex& temp = *this = z
  cout << endl << endl; </p>
// 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;
                 real
                        14
                                                   14
                                            real
             a
                                    temp1
                        15
                 imag
                                            imag
                                                   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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            Output
                                            (4,5)
      real += x.real; imag += x.imag;
                                                      this
                                            (4,5)
      return this;
                                            (4,5)
                                                                        14
                                                                 real
                                                         temp1
    Complex& add3(const Complex& x)
                                            (14,15)
                                                                 imag
                                                                        15
    // 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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // 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)
                                                                 imag
                                                                        15
    // 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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
      return (*this);
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // 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)
                                                                 imag
                                                                        25
    // 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
                                                                 X
                                                                             10
                                                                      real
                                                                 W
                                                                     imag
                                                                             10
      real += x.real; imag += x.imag;
      return (*this);
                        Returning it by value, we need to construct a
                        temporary object "temp2" which is a copy of temp1
                                                                             14
                                                                      real
    Complex* add2(const Complex& x)
                                                                 a
    // 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)
                                                                     imag
                                                                             25
    // Return by reference
                                               (14,15)
      real += x.real; imag += x.imag;
                                                                             24
                                                                     real
      return (*this);
                                                             temp2
                                                                     imag
                                                                             25
};
```

```
#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
                                                                real
                                              (14,15)
                                                                imag
                                                                       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;
                 real
                        14
                                                  24
                                                                       24
                                                                real
                                            real
             a
                                    temp1
                                                        temp2
                        15
                 imag
                                                                imag
                                                  25
                                                                       25
                                            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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
      return this;
                                            (24,25)
                                                                        24
                                                                 real
                                                         temp1
    Complex& add3(const Complex& x)
                                                                 imag
                                                                        25
    // Return by reference
                                                      this
      real += x.real; imag += x.imag;
                                                                        24
                                                                 real
      return (*this);
                                                         temp2
                                                                 imag
                                                                        25
};
```

```
#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
                                                                  real
                                               (14,15)
                                                                  imag
                                                                        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, This is temp1 | t << This is temp2 endl;
  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 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;
                 real
                        14
                                             real
             a
                                    temp:
                                                         temp2
                        15
                 imag
                                                                  imag
                                                                        25
                                             irnag
```

```
#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
                                                                real
                                              (14,15)
                                                                imag
                                                                       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;
  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
                       14
             a
                 imag
                        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
                                                                        10
                                                                 real
                                                            W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                                      this
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
    // Return by value using pointer
                                                                 imag
                                                                        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);
};
```

```
#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)
                                  Cursor here
int main()
                                                                       10
                                                                real
                                              (24,25)
                                                                imag
                                                                       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;
                 real
                        14
             a
                 imag
                        15
```

```
#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)
                                                                imag
                                                                       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;
}
                 real
                        14
                                  real
                              b
             a
                 imag
                        15
                                  imag
```

```
class Complex /* File: complex.h */
                                            r
                                                    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
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                 real
      return this;
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                         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
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
      return this;
                                                                 real
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        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;
                                              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)
                                                                imag
                                                                       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;
}
                 real
                        14
                                  real
                              b
             a
                 imag
                        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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                 real
      return this;
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        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
                                                             Χ
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        14
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                         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
                                                             Χ
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        14
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        15
    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
                                                             Χ
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        14
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        15
    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;
                                              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)
                                                                imag
                                                                       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); Complex w(10, 10); Complex w(10, 10);
  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 << end1;
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << endl;
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0;
}
                 real
                        14
                                  real
                                         14
                              b
             a
                 imag
                        15
                                  imag
                                         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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        14
                                                                 real
      return this;
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        15
    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
                                                             Χ
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        24
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        15
    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
                                                             Χ
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        24
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    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
                                                             Χ
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        24
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    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;
                                              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)
                                                                imag
                                                                       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); Complex w(10, 10); Complex w(10, 10);
  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;
  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
                        14
                                   real
                                         24
                              b
             a
                                         25
                 imag
                        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
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                      this
                                                                        24
                                                                 real
      return this;
                                                             h
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
      real += x.real; imag += x.imag;
      return (*this);
```

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

```
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)
                                      Cursor here 、
                                                  (14,15)
int main()
                                                                       10
                                                                 real
{
                                                  (24,25)
                                                                imag
                                                                       10
                                                  (24,25)
  // 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
                        14
                                   real
                                          24
                              b
             a
                 imag
                        15
                                   imag
                                          25
```

```
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
                                                  (24,25)
                                                                 imag
                                                                        10
                                                  (24,25)
  // 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
                        14
                                   real
                                          24
                                                    real
                              b
             a
                                                C
                 imag
                        15
                                   imag
                                          25
                                                    imag
```

```
class Complex /* File: complex.h */
                                             r
                                                    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
                                                                  real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                  real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                  real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                  imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                       this
                                                                         4
                                                                  real
      real += x.real; imag += x.imag;
                                                                  imag
                                                                         5
      return (*this);
};
```

```
class Complex /* File: complex.h */
                                             r
  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
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                         4
                                                                 real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                         5
      return (*this);
```

```
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
                                                  (24,25)
                                                                 imag
                                                                        10
                                                  (24,25)
  // 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
                        14
                                   real
                                          24
                                                    real
                              b
             a
                                                C
                 imag
                        15
                                   imag
                                          25
                                                    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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                         4
                                                                 real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                         5
      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
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                        14
                                                                 real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                         5
      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
                                                                  real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                  real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                  real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                        14
                                                                  real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                        15
      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
                                                                    real
                                                               W
                                              Output
                                                                    imag
                                                                           10
      real += x.real; imag += x.imag;
                                              (4,5)
      return (*this);
                                              (4,5)
                                              (4,5)
                                                                           14
                                                                    real
    Complex* add2(const Complex& x)
                                                               a
    // Return by value using pointer
                                                                    imag
                                                                           15
                                              (14,15)
                                              (14,15)
                                              (14,15)
      real += x.real; imag += x.imag;
                                                                           24
      return this;
                                                                    real
                                                               b
                                              (24,25)
                                              (14,15)
                                                                    imag
                                                                           25
    Complex& add3(const Complex& x)
                                              (24,25)
    // Return by reference
                                              (24,25)
                                                        this
                                                                           14
                                                                    real
      real += x.real; imag += x.imag;
                                                                    imag
                                                                           15
      return (*this);
              Returning it by reference, i.e. returning of
};
```

```
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
                                                  (24,25)
                                                                imag
                                                                       10
                                                  (24,25)
  // 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. This is c_1)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add^2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.ad d3(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 << end1;
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0;
}
                 real
                        14
                                   real
                                         24
                                                           14
                                                    real
                              b
             a
                                               C
                                   imag
                 imag
                        15
                                          25
                                                    imag
                                                           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
                                                             X
                                                                        10
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                        14
                                                                 real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                        15
      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
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                        24
                                                                 real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                        15
      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
                                                                 real
                                                             W
                                            Output
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                        24
                                                                 real
      real += x.real; imag += x.imag;
                                                                 imag
                                                                        25
      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
                                                                    real
                                                               W
                                              Output
                                                                   imag
                                                                           10
      real += x.real; imag += x.imag;
                                              (4,5)
      return (*this);
                                              (4,5)
                                              (4,5)
                                                                           14
                                                                    real
    Complex* add2(const Complex& x)
                                                               a
    // Return by value using pointer
                                                                   imag
                                                                           15
                                              (14,15)
                                              (14,15)
                                              (14,15)
      real += x.real; imag += x.imag;
                                                                           24
      return this;
                                                                    real
                                                               b
                                              (24,25)
                                              (14,15)
                                                                    imag
                                                                           25
    Complex& add3(const Complex& x)
                                              (24,25)
    // Return by reference
                                              (24,25)
                                                        this
                                                                           24
                                                                    real
      real += x.real; imag += x.imag;
                                                                    imag
                                                                           25
      return (*this);
              Returning it by reference, i.e. returning of
};
```

```
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
                                                  (24,25)
                                                                imag
                                                                       10
                                                  (24,25)
  // 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. This is c_1)).print(); // Complex temp = *this = x
  Complex y(4, 5); (y.add^2(w))->print(); // Complex* temp = this = &y
  Complex z(4, 5); (z.add(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 << end1;
  Complex b(4, 5); b.add2(w)->add2(w)->print(); b.print(); cout << end1;
  Complex c(4, 5); c.add3(w).add3(w).print(); c.print();
  return 0;
}
                 real
                        14
                                   real
                                         24
                                                           24
                                                    real
                              b
             a
                                               C
                                   imag
                 imag
                        15
                                         25
                                                    imag
                                                           25
```

```
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
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                  real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                  real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                  imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                       this
                                                                        24
                                                                  real
                                            ( 24 , 25 )
      real += x.real; imag += x.imag;
                                                                  imag
                                                                        25
      return (*this);
```

**}**;

```
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()
                                                                 real
                                                                        10
                                                  (24,25)
                                                                 imag
                                                                        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
  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
                        14
                                   real
                                          24
                                                           24
                                                    real
                              b
             a
                                                C
                                   imag
                 imag
                        15
                                          25
                                                    imag
                                                           25
```

```
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
                                                                 imag
                                                                        10
      real += x.real; imag += x.imag;
                                            (4,5)
      return (*this);
                                            (4,5)
                                            (4,5)
                                                                        14
                                                                 real
    Complex* add2(const Complex& x)
                                                             a
    // Return by value using pointer
                                                                 imag
                                                                        15
                                            (14,15)
                                            (14,15)
                                            (14,15)
      real += x.real; imag += x.imag;
                                                                        24
      return this;
                                                                 real
                                                             b
                                            (24,25)
                                            (14,15)
                                                                 imag
                                                                        25
    Complex& add3(const Complex& x)
                                            (24,25)
    // Return by reference
                                            (24,25)
                                                      this
                                                                        24
                                                                 real
      real += x.real; imag += x.imag;
                                            (24,25)
                                            (24, 25)
                                                                 imag
                                                                        25
      return (*this);
```

**}**;

```
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()
                                                                 real
                                                                        10
{
                                                  (24,25)
                                                                 imag
                                                                        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;</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
                        14
                                   real
                                          24
                                                           24
                                                    real
                              b
             a
                                                C
                 imag
                        15
                                   imag
                                          25
                                                    imag
```

```
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
                                                  (24,25)
                                                                imag
                                                                       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;</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
                        14
                              b
             a
                        15
                 imag
                                   imag
```

```
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
                                                  (24,25)
                                                                imag
                                                                       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;</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
             a
```

```
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()
                                                                real
                                                                      10
                                                 (24,25)
                                                                imag
                                                                      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;
```

```
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()
                                                                real
                                                                      10
                                                 (24,25)
                                                                imag
                                                                      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 🗶
```

```
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()
                                                                real
                                                                      10
                                                 (24,25)
                                                                imag
                                                                      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;
```

```
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;
  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;
```

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()
                                                 (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;
  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;
  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;
```