

1	2	3	4	5	6	7	8	9	10

1. Для класса комплексных чисел `template<class T> class complex { T r; T i; ... };` написать реализацию операции "==" , выполняющей сравнение двух комплексных чисел методом класса и функцией-другом класса.

2. Объект x принадлежит классу X , для которого определены конструктор преобразования типа `int` в класс X и функция преобразования класса X в тип `int`. Описать прототипы двух перегруженных функций $f()$ из некоторой области видимости, для которых будут верны следующие обращения к ним:

```
f ("p");    f (x, 0);    f (0, 0);    f (x, "q");    f (1, "r");
```

3. Есть ли ошибки в реализации функции `main()`? Удалите неверные операторы из этой функции (если они есть). Для всех правильных операторов этой функции, используя операцию разрешения области видимости "::", также укажите, из какой области видимости вызываются используемые в них функции $f()$.

```
int x = 0;
int f (int a) { return x = a; }
class A { int x;
public: A (int n = 1)          { x = n; }
      int f (int a)          { return x = a; } };
class B { int x;
public: B (int n = 2)          { x = n; }
      virtual int f ()        { return x = ::x; } };
class C: public A, public B { int x;
public: int f (int a) { return x = a; }
      int f ()          { return x = ::x; } };
int main () { A a; B b; C c;
             A * pa = & a; B * pb = & b; C * pc = & c;
             x =      f (3); x = pa -> f (); x = pa -> f (4);
             pa = pc; x = pa -> f (); x = pa -> f (5); x = pb -> f ();
             x = pb -> f (6);
             pb = pc; x = pb -> f (); x = pb -> f (7); x = pc -> f ();
             x = pc -> f (8);
             return -1;
}
```

4. Написать функцию $g()$ с тремя параметрами: непустой и неизменяемый контейнер-список типа `list<long int>`, непустой контейнер-вектор типа `vector<long int>`, целое число – шаг по второму контейнеру. Функция должна копировать отрицательные элементы списка с шагом, равным 1, в уже имеющийся контейнер-вектор, от его начала к концу с шагом, равным третьему параметру, а затем распечатывать элементы вектора в прямом порядке. Функция возвращает количество измененных элементов вектора.

5. Описать особенности процесса отождествления типов параметров перегружаемых функций с несколькими параметрами. Приведите по одному примеру перегрузки функции с несколькими параметрами, в которых

- не удастся найти ни одной подходящей функции,
- находится одна подходящая функция,
- находится несколько подходящих функций.

6. Для объектов из задания 3 определить, тела каких конструкторов и деструкторов (возможно пустые, если они не определены явно) и в каком порядке будут исполнены при работе следующего фрагмента программы:

```
int main () { struct D { A a; C c; }; D d; }
```

7. Есть ли ошибки в приведенном фрагменте программы на C++? Если есть, то **объясните**, в чем они заключаются. Ошибочные конструкции вычеркнуть из текста программы. Что будет выдано в стандартный поток вывода при работе программы?

```
class C {
public: virtual void f (int x) { h (x); cout << "C::f," << x << endl; }
       virtual void g ()      { h (0); cout << "C::g"      << endl; }
       virtual void h ()      {      cout << "C::h"      << endl; }
       virtual void h (int k) { h ();   cout << "C::h," << k << endl; }
};
class D: public C {
public: virtual void f (int y) { h (y); cout << "D::f," << y << endl; }
       virtual void g ()      { h (1); cout << "D::g"      << endl; }
       virtual void h ()      {      cout << "D::h"      << endl; }
       virtual void h (int k) { h ();   cout << "D::h," << k << endl; }
};
int main( ){ C c; D d; C * p = & d;
            p -> f (2); p -> g (); p -> h (); p -> h (3); }
```

8. Дать определение полиморфизма. Описать различные виды полиморфизма в Си++. Привести необходимые примеры фрагментов программ.

9. Что будет выдано в стандартный канал вывода при работе следующей программы?

```
struct X; void f(X & x, int n);
int const P = 1; int const Q = 1; int const R = 1;
struct X {
    X()      { try { f(*this, 0); cout << 1 << endl; }
              catch (X)      { cout << 2 << endl; }
              catch (int)    { cout << 3 << endl; }
    }
    X (X &)      { cout << 4 << endl; }
    ~X ()       { cout << 5 << endl; }
};
struct Y: X { Y ()      { f(*this, -1); cout << 6 << endl; }
              Y (Y &)   { cout << 7 << endl; }
              ~Y ()    { cout << 8 << endl; }
};
void f(X & x, int n) { try { if (n < 0) throw x;
                           if (n > 0) throw 1;
                           cout << 9 << endl;
                           }
                  catch (int)      { cout << 10 << endl; }
                  catch (X& a)     { cout << 11 << endl;
                                    f(a, 1);
                                    cout << 12 << endl;
                                    throw;
                                  }
}
int main() { try { Y a; }
            catch (...) { cout << 13 << endl;
                          return 0;
                        }
            cout << 14 << endl;
            return 0;
}
```

10. Есть ли синтаксические ошибки в тексте приведенной программы? Если можно исправить описание класса, не вводя дополнительных членов, чтобы программа стала верной, то как?

```
class A { static int i;
         void f () { if (i >= 0) i = -1, g ();
                    cout << "f ()" << endl; }
         void g () { f (); cout << "g ()" << endl; }
};
int A::i = 1;
int main () { A::i = 1; const A a; a.f(); a.i = 0; }
```