Вариант 3_2009 Ф.И.О._____№ группы_____

1	2	3	4	5	6	7	8	9	10

1. 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 T {
     public: virtual void f (int x) { h ();  cout << "T::f," << x << endl; }
                     void g ()      { h ();  cout << "T::g"       << endl; }
             virtual void h ()      {        cout << "T::h"       << endl; }
};
class U: virtual public T {
     public:         void f (int y) { h (y); cout << "U::f," << y << endl; }
             virtual void g ()      { h (0); cout << "U::g"       << endl; }
                     void h (int k) {        cout << "U::h," << k << endl; }
};
int main( ){ T t; U u; T * p = & u;
             p -> f (1);  p -> g ();  p -> h ();  p -> h (2); }
```

2. Есть ли ошибки в реализации функций *B::g()* и *main()*? Если есть, исправьте неверные операторы из этой функции, используя операцию разрешения области видимости "**::**".Для всех операторов этих функций укажите, из какой области видимости вызываются используемые в них функций *f()* и *g()*.

```
int x = 0;
int f ()                                  { return x = 1;     }
class A { int x;
          public:A ( int n = 2)           {        x = n;     }
                    int f ()              { return x = 3;     }
                    int f (int a, int b) { return x = a % b; }
};
class B:  public A { int x;
          public:    int f (int a)        { return x = a;     }
                     int f (int a, int b) { return x = a + b; }
                     int g (A * a, B * b);
};
int  B::g (A * pa, B * pb)
{                   x =       f ();
                    x =       f (5);
                    x =       f (6, 6);
                    x =    A::f (5);
                    return -1;
}
int main () { B a; x =     a.f ();
                    x =     a.f (7);
                    return  a.g (& a, & a);
}
```

3. Привести по одному примеру перегрузки бинарной, унарной префиксной и постфиксной операций (всего 3 примера). Указать особенности перегрузки операций присваивания. Можно ли изменить приоритет операции при её перегрузке?

4. Для класса комплексных чисел `template<class T> class complex { T r; T i; ... };` написать два варианта реализации префиксной операции увеличения "++", выполняющей увеличение на 1 действительной части комплексного числа методом класса и функцией-другом класса.

5. Данные каких типов могут подвергаться динамическому и статическому приведению? В чём проявляется отличие выполнения динамического приведения типов для указателей и ссылок? Какие свойства указателей и ссылок влияют на это отличие? Приведите примеры фрагментов программы с использованием подобных операций.

6. Написать функцию *g()* с тремя параметрами: непустой и неизменяемый контейнер-вектор типа *vector<double>*, непустой контейнер-список типа *list<double>*, целое число – шаг по первому контейнеру. Функция должна сравнивать элементы списка, выбираемыми от его начала с шагом, равным 1, с элементами вектора, выбираемыми от начала с шагом, равным третьему параметру. Если обнаруживается несовпадение очередной выбранной пары, то в список в текущем месте вставляется отсутствующий элемент. Изменённый список распечатывается в обратном порядке. Функция возвращает количество элементов, вставленных в список.

7. Для объектов из задания **2** 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 () { A a; class C { B b; A a; public: C (): b(), a (b) {} }; C c; }
```

8. Описать прототипы двух перегруженных функций *f()* из некоторой области видимости, для которых будут верны следующие обращения к ним:

```
f (1000000000000);  f (1);   f ();   f (0, 0);   f ("t");   f (1, "u");
```

9. 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 () const { if (i < 0) g (i);
                                    cout << "f ()" << endl; }
          void g (int & n) { i = n; f (); cout << "g ()" << endl; }
};
int A::i = 1;
int main () { const A a; a.g (2); }
```

10. 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, -1);  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, 0);
                                                cout << 12 << endl;
                                                throw;
                                              }
}
int main() {          try { Y a; }
                      catch (...)             { cout << 13 << endl;
                                                return 0;
                                              }
                                                cout << 14 << endl;
                                                return 0;
}
```