

1	2	3	4	5	6	7	8	9	10

NB! Во всех задачах, где это требуется, предполагается наличие необходимых включений и директивы `using namespace std`.

1. Для каждого имени из реализации функций `B::g()` и `main()` укажите (используя операцию `::`) область видимости имён так, чтобы программа была верной. **Подчеркните** вставленные расширения имён, без которых программа ошибочна. Если вариантов расширения имён несколько, укажите все.

```
static int x;
namespace N1 { int f (int a, int b){ return x = a + b; }
class A { int x;
public: A (int n = 13) { x = n; }
int f(){ return ::x = x; }
};
namespace N2 { class B: public A { float x;
public: int f (int a) { return ::x = x; }
void g ();
};
}
}
void B::g() { x = 20.13; f ();
f (2013); f (13, 20132013);
}
int main () { B b; b. f (); b. f (13);
x = f ('2', 013);
return 0;
}
```

2. Есть ли ошибки в функции `main()` приведённого фрагмента на Си++? Если есть, то **объясните**, в чём они заключаются. Если нужно, удалите ошибочные вызовы из функции `main()`. Что будет выдано в стандартный поток вывода при работе исправленной (если нужно) программы?

```
class A { public: virtual void f() { h (65); cout << "A::f" << endl; }
virtual void g(int x) { h (x); cout << "A::g," << x << endl; }
void h(int x) { cout << "A::h," << x << endl; }
};
class B: public A { public:
void g (int x) { h (x); cout << "B::g," << x << endl; }
virtual void h (int x) { cout << "B::h," << x << endl; }
};
class C: public B { float f;
public: void g () { h (67); cout << "C::g" << endl; }
virtual void h (int x) { cout << "C::h," << x << endl; }
};
int main() {
C c, * pc = & c; B * pb = & c; A * pa = & c;
pa -> f (); pa -> f (2013);
pa -> g (13); pb -> f ();
pb -> g (); pb -> g (13);
pb -> h (0); pc -> f ();
pc -> g (); pc -> h ();
pc -> h (0); return 1;
}
```

3. Что такое функция преобразования в языке Си++? Какие ограничения накладываются на использование функций преобразования при выборе одной из перегруженных функций? Привести пример использования функции преобразования в программах на Си++.

4. Используя описания классов А и В из задания 1, определить, тела каких конструкторов и деструкторов (возможно явно не определённые) и в каком порядке будут исполнены при работе следующего фрагмента программы, расположенного в пространстве имён `N2` из задания 1:

```
void h () { struct C { A a; B b; }; A a(2013); C c; }
```

5. Для класса рациональных дробей

```
template<class T> class fraction { T n; T d; ... };
```

написать два варианта реализации вне класса операции "/", выполняющей деление одной рациональной дроби на другую, а именно – методом класса и функцией-другом этого класса.

6. Описать алгоритм определения возможности использования имён из базовых классов в производном классе в случае множественного наследования. Привести по одному примеру, демонстрирующему случаи обнаружения ошибки на каждом из шагов алгоритма.

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

```
struct X; void f(X & x, int n);
struct X { X () { cout << "Xc";
              f (*this, -1); cout << "Xf"; }
          X (X &) { cout << "Xk"; }
};
struct Y: X { Y () { try { cout << "Yt"; f (*this, 1);
                        cout << "Yp";
                    }
                catch (Y) { cout << "Yy"; throw; }
                catch (int) { cout << "Yi"; throw; }
            }
          Y (Y &) { cout << "Yk"; }
};
void f (X & x, int n) { try { cout << "ft";
                            if (n < 0) throw x;
                            cout << "f0";
                            if (n > 0) throw 1;
                            cout << "f1";
                        catch (int) { cout << "fi"; throw; }
                        catch (X& a) { cout << "fc";
                                        f(x, 1);
                                        cout << "fr";
                                        throw;
                        }
                    }
int main() { try { Y a; } catch (...) { cout << "mc"; return 0; }
            cout << "mr";
            return 0;
        }
}
```

8. В приведённой программе возможно наличие синтаксических ошибок в определении класса A. Если ошибки есть, исправьте их заменой, исключением или добавлением нужных служебных слов языка Си++. Обоснуйте сделанные исправления.

```
class A { int i;
          int f (int & x) { return g (x); }
          int g (int & x) { if (x >= 0) f (-i); return i; } };
int A::i = 2013;
int main () { const A a; A::i = 201; a.f (20); return a.i = 1; }
```

9. Для приведённой ниже программы описать функцию $B^* f(A^*)$, которая, получая в качестве параметра указатель типа A^* , возвращает его значение, наиболее безопасным образом преобразованное к типу B^* , а в случае невозможности преобразования корректно завершает работу программы.

```
struct A { };
struct B:A {int x; B (int y = 5){ x = y; } };
int main () { try { B b, *pb = f(&b); cout << pb -> x << '\n';
                  A a; pb = f(&a); cout << pb -> x << '\n'; }
            catch (...) { } return 0;
        }
}
```

10. Даны описания: `typedef vector<int> V;`
`struct Weight_t { V::size_type Index; // индекс элемента вектора`
`float Weight; }; // вес элемента вектора`
`typedef list <Weight_t> L;`

Написать функцию $g()$, которая по заданному вектору типа V и соответствующему ему списку типа L вычисляет средневзвешенное значение элементов вектора (средний результат умножения элементов на их веса), выдавая в выходной поток значения и веса элементов.