

Контрольная работа C++. Вариант 1.

1. Что будет напечатано данной программой?

```
struct A {
    A( int i = 0 ){cout<<"A(int) ";}
    A( const A&a ){ cout<<"A(const-A&) ";}
    A( A& a ){cout<<"A(A&) ";}
    ~A(){cout<<"~A ";}
    operator int() const { cout<<" op-int "; return 1;}
};
struct B : A {
    B( const B& b){ cout<<"B(const-B&) ";}
    B( int i=0) : A( *this ){ cout<<"B(int) ";}
    ~B(){cout<<"~B ";}
};
int main(){
    { const A a1 = 1; cout<<endl;
      B b (a1); cout<<endl;
      A a2=a1;
    }
    {cout<<endl; B b(1);}
    return 0;
}
```

2. Напишите функцию **C::f()** так, чтобы она возвращала в качестве результата сумму всех полей объекта C и результатов вызова всех его функций-членов (кроме C::f()). Уточнение области видимости имен используйте **только там, где это необходимо**. Если в функции **main** есть ошибки, исправьте их.

```
struct Base {      int a;
                  int f(int x){return a*x;}
};
struct B1 : virtual Base { int b; };
struct A1 : Base {int a; };
struct B2 : virtual Base { int a; };
struct A2 : Base { float f; double d;};
struct C : A1,A2, B1, B2 { int a; int f(); };
int C::f(){ return (.....); }

namespace N { int a = 1;
              int f(){return 2;}};
using namespace N;

int main(){ int a = 5; int f = 5;
C c;
/*тут инициализация полей объекта с ..... */
cout<< c.f() + a + f + f();
return 0;
}
```


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

```
int f(int i);
struct A { A (int i=0) { cout << "A_Constr\n"; f(i); }
          A (const A & a) { cout << "A_Copy\n"; }
          ~A () { cout << "A_Destr\n"; }
};
struct B : A { B () { cout << "B_Constr\n"; }
             B (const B & a):A(a) { cout << "B_Copy\n"; }
             ~B () { cout << "B_Destr\n"; }
};
void g(A & a) { throw a;}

int f(int i){ try{ if (i>=0 && i<10)return i*i;
                 if(i<20) throw 1;
                 else { B b; throw b;}
               }
             catch(int){cout<<" f_int_catch\n";}
             catch(A& a){cout<<" f_A&_catch\n"; throw;}
}

int main () {
    try {
        A a1(15), a2(25); g(a1);
    }
    catch ( B& ) { cout << "B&_Catch\n"; }
    catch ( A& ) { cout << "A&_Catch\n"; }
    return 0;
}
```

3. Напишите 2 класса A и B, A и B должны быть абстрактными типами данных. Напишите внешнюю функцию count, которая выводит на экран общее количество объектов обоих классов, существующих на момент вызова функции. Глобальные переменные использовать запрещено. Должна выполняться следующая функция main().

```
int main(){
    count(); //0
    A a_arr[4];
        { B b_arr[5];
          count(); //9
          A* pa = b_arr;
        }
    count(); //4
    return 0;
}
```

4. Если в данной программе есть ошибки, объясните, в чем они состоят, ошибочные операторы вычеркните. Что будет напечатано в результате работы функции main?

```
#include <iostream>
using namespace std;
class A {
public: int f(int x){ h(); cout<<"A::f(int) "; return 0;}
virtual int g(int x){ f(x);cout<<"A::g(int) "; return 0;}
virtual void h(){ cout<<"A::h() "; return;}
};
class B:public A {
public: int f(){ h(); cout<<"B::f() ";return 0;}
        int g(int x){f(); cout<<"B::g(int) ";return 0;}
        virtual void h(){ cout<<"B::h() "; return;}
};
int main(){
    A* p; B b; p=&b;
    p->f(1);    p->f(); p->g(1);    p->h();
    return 0;
}
```