

Object oriented programming

Task1:	Task2:
Task3:	Task4:
Task5:	Task6:
Task7:	

Task1	Task2	Task3	Task4	Task5	Task6	Task7	Sum

Task 1: What is the output of the following program?

```
#include <iostream>
using namespace std;

class A
{
public:
    int no;
public:
    A( ) { no = 4;};

    void f(A* x) {
        no += x->no;
    };
    A* get( ) { return this;}
};

class B : public A
{
public:
    B( ) {
        no++;
        A::f(this);
    };
    virtual void f(A* x) {
        no += x->no;
    };
};

class C : public A
{
public:
    C( ) {
        B* b = new B;
        b->f(b);
        no += b->no;
    }
    void f(A* x) { no -= x->no + 3; };
};

class D : private B, public C
{
public:
    D( ) {
        C* c = new C;
        f(c);
    };
    void f(A* x) {
        C::f(C::get());
        C::no += (x->no + B::no);
        cout << C::no;
    };
};

void main( ) {
    C* d = new D;
    d->f(d);
}
```

Task2: What is the output of the following program?

```
#include <iostream>

using namespace std;

int data = 0;

class A
{
public:
    A( ){data++;}
    A& operator++( ) { cout << "1 " << data-- << endl; A* b = new A; return *b;}
    A operator++(int) { cout << "2 " << data++ << endl; A* b = new A; return *b;}
    A& operator--( ) { cout << "3 " << ++data << endl; A* b = new A; return *b;}
    A operator--(int) { cout << "4 " << --data << endl; A* b = new A; return *b;}
    void operator+(A& b) { data++;}
    ~A( ){data++;}
};

class B
{
public:
    virtual void f(int d = 2) = 0;
};

class C : public A, public B
{
public:
    void f(int d = 5) { A a; a++; data+=d--;;}
};

void main()
{
    A i, j;

    B *c = new C;
    --++i--++j--;
    c->f(data);
}
```

Task 3: What is the output of the following program?

```
#include <iostream>

using namespace std;

int id = 0;

class CreateAndDestroy {
public:
    CreateAndDestroy() { id = 1;}
    CreateAndDestroy(int objectNumber) {
        objectID = objectNumber;
        cout << objectID << " constructor" << endl;
    }

    ~CreateAndDestroy() {
        cout << id << " destructor" << endl;
    }
private:
    int objectID;
};

void create( void );

CreateAndDestroy a(id++);

class CD1 : virtual public CreateAndDestroy {
public:
    CD1( int i): CreateAndDestroy(i) {CreateAndDestroy b(i);}
    ~CD1() {CreateAndDestroy b(id++); }
};

class CD2 : virtual public CreateAndDestroy {
    CreateAndDestroy a;

public:
    CD2(int i) : CreateAndDestroy(i) {}
    virtual ~CD2() { CreateAndDestroy b(id++); }
};

class CD3 : virtual public CD1, public CD2 {
public:
    CD3(int i) : CD1(i), CD2(i) { static CreateAndDestroy a(i);}
};

void main( ) {
    CreateAndDestroy b(id++);
    static CreateAndDestroy c(id++);
    CD3 a(id);
    create();
    CreateAndDestroy f(id++);
}

void create( void ) {
    CreateAndDestroy a(id++);
    static CreateAndDestroy b(id++);
    CreateAndDestroy c(id++);
}
```

Task 4: What is the output of the following program?

```
#include <iostream>
#include <complex>

using namespace std;

class Base {
public:
    virtual void f( int );
    virtual void f( double );
    virtual void g( int i = 10 );
};

void Base::f( int ) { cout << "Base::f(int)" << endl; }

void Base::f( double ) { cout << "Base::f(double)" << endl; }

void Base::g( int i ) {
    cout << i << endl;
    if (i==10) { f(2*i); }
}

class Derived: public Base {
public:
    void f( complex<double> );
    void g( int i = 20 );
};

void Derived::f( complex<double> ) { cout << "Derived::f(complex)" << endl; }

void Derived::g( int i ) {
    cout << "Derived::g() " << i << endl;
    if (i==10) { f(2*i); }
}

void f(Base &a) { a.g(); }

void main() {
    Base    b;
    Derived d;
    Base*   pb = new Derived;
    b.f(1.0);
    d.f(1.0);
    pb->f(1.0);
    b.g();
    d.g();
    pb->g();
    f(d);
    delete pb;
}
```

Task 5: What is the output of the following program?

```
#include <iostream>
using namespace std;

class A
{
public:
    float x;

public:
    A(): x(1){}
    void f(int i) { x+=i; cout << "asd";}
    virtual void f(float i) { x*=i; cout << "aca"; }
    void f1(int i) {x*=i+2;}
    void f1(int i, float j) { x*=i*j; cout << "bcs" << endl;}
    virtual void f1(float i, float j) {x-=i*j; cout << "ort" << endl;}

};

class B : public A
{
public:
    void f(float) { f1(2,3); x = x * 10; cout << "swe" ;}
    void f1(float i, float j = 4) {x-=i*j; cout << "otk" ;}

};

void main()
{
    A a;
    B b;
    a.f(a.x);
    b.f1(2,a.x);
    b.f(1);
    b.f1(1);
    cout << a.x << endl;
    cout << b.x << endl;
}
```

Task 6: What is the output of the following program?

```
#include <iostream>

using namespace std;

class My {
public:
    My(){}
};

class Dy : public My {
public:
    Dy(){}
};

void f(int i) {
    switch (i) {
        case 1: throw 1.2f; break;
        case 2: throw 3.2; break;
        case 3: throw My(); break;
        case 4: throw Dy(); break;
    }
}

void g(int i) {
    switch (i) {
        case 0: throw 2.3; break;
        case 5: throw 3.2f; break;
        case 6: throw My(); break;
        case 7: throw Dy(); break;
    }
}

void main() {
    for(int i=0; i<8; i++) {
        try {
            f(i);

            try {
                g(i);
            }
            catch (Dy) {cout << "Alo";}
            catch (My) {cout << "Hey";}
            catch (double) {cout << "Uau";}
            catch (float ) {cout << "Kss";}
        }
        catch (My) {cout << "Uau";}
        catch (float ) {cout << "Alo";}
        catch (Dy) {cout << "Kss";}
        catch (double ) {cout << "Hey";}

        cout << i << endl;
    }
}
```

Task 7: What is the output of the following program?

```
#include <iostream>
using namespace std;

class A {
public:
    virtual int addition (int a, int b) { cout << "A"; return (a+b); }
    int subtraction (int a, int b) { cout << "B";return (a-b); }

};

class B : public A {
public:
    int addition (int a, int b) { cout << "D"; return (a+2*b); }
    virtual int subtraction (int a, int b) { cout << "E";return (a-b); }

};

int main ()
{
    A *a = new B();
    B *b = new B();
    cout << a->addition(a->subtraction(5, 4), b->addition( a->addition(1,2),
b->subtraction(3,2))) << endl;
    return 0;
}
```


Rešenja i način bodovanja:

Task1:

31

10 poena

Task2:

4 2

4 2

2 3

1 5

3 6

2 12

10 poena integralno

Task 3:

0 constructor

1 constructor

2 constructor

3 constructor

1 destructor

3 constructor

1 constructor

2 constructor

3 constructor

4 destructor

4 destructor

4 constructor

5 destructor

5 constructor

6 destructor

6 destructor

6 constructor

7 destructor

7 destructor

7 destructor

7 destructor

7 destructor

7 destructor (VS2008 ima još jedan red 7 destructor)

10 poena integralno

Task 4:

Base::f(double)
Derived::f(complex)
Base::f(double)
10
Base::f(int)
Derived::g() 20
Derived::g() 10
Derived::f(complex)
Derived::g() 10
Derived::f(complex)

10 poena integralno

Task 5:

acaotkotsweotk1
-74

10 poena integralno

Task 6:

Uau0
Alo1
Hey2
Uau3
Uau4
Kss5
Hey6
Alo7

10 poena integralno

Task 7:

EDDBD15

10 poena integralno