

## 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;

unsigned long address = 0;

int dword = 8;
int dd = 7;
int word = 4;
int byte = 1;

class A
{
public:
    unsigned long operator+() {cout << "operator+()" << endl;
        return address+word; }
    unsigned long operator+(unsigned long) { cout << "operator+(ul)" << endl;
        return address+word; }
    unsigned long operator+(A) { cout << "operator+(A)" << endl;
        return address+dd; }
    unsigned long operator-() {cout << "operator-()" << endl;
        return address-dword; }
    unsigned long operator-(unsigned long) {cout << "operator-(ul)" << endl;
        return address-word; }
    unsigned long operator-(A) {cout << "operator-(A)" << endl;
        return address-dd; }
};

    unsigned long operator+(A a) {cout << "operator+(A)ext" << endl;
        return address+dd+6 ;}
    unsigned long operator+ (unsigned long g, A a)
        {cout << "operator+(ul,A)ext" << endl; return g+6; }
    unsigned long operator- (A a) {cout << "operator-(A)ext" << endl;
        return address-dd-7; }
    unsigned long operator-(unsigned long g, A a)
        {cout << "operator-(g,A)ext" << endl ; return g-7; }

int main(int argc , char* argv[])
{
    A a, d, e, m, n;
    unsigned long b, c, f;
    address = 3000;
    b = 3U; c = 5U; f = 1U;
        cout << a+d-b-e+c-f+(n+m)-m << endl;
    return 0 ;
}
```

## **Task2: What is the output of the following program?**

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

class Node
{
protected:
    struct bbox_struct {
        double Xmin , Ymin, Xmax, Ymax;
    } bbox;
    char data[10];
    int len;
public:
    Node();
    void SetBBox(double x1, double y1, double x2, double y2);
    bool Insert(double x1, double y1, double x2, double y2, char ch);
    void Draw();
};

Node::Node() {data[0]=0; len=0;}
void Node::SetBBox(double x1, double y1, double x2, double y2)
{
    bbox.Xmin=x1; bbox.Ymin=y1; bbox.Xmax=x2; bbox.Ymax=y2;
}

bool Node::Insert(double x1, double y1, double x2, double y2, char ch)
{
    if(bbox.Xmin<x1 && bbox.Xmax>x2 && bbox.Ymin<y1 && bbox.Ymax>y2)
    {
        data[len++]=ch; data[len]=0;
        return true;
    }
    else
        return false;
}

void Node:: Draw() { if(data[0]) cout << bbox.Xmin << " " << bbox.Ymin << " " <<
    bbox.Xmax << " " << bbox.Ymax << " " << data << endl ; }

class Tree
{
protected:
    int nodeNo, level;
    Node* vect;
public:
    Tree (int lev);
    ~Tree();
    bool Insert(int i, double x1, double y1, double x2, double y2, char ch);
    void SetBBox(int i, double x1, double y1, double x2, double y2);
    void Draw(int i);
};

Tree::Tree(int lev)
{
    level = lev;
    nodeNo = 1;
    int factor = 4;
    for(int i=1; i<lev; i++)
    {
        nodeNo += factor;
        factor *= 4;
    }
    vect = new Node[nodeNo];
}
```

```

Tree::~Tree() {delete[] vect;}
bool Tree::Insert(int i, double x1, double y1, double x2, double y2, char ch)
{
    bool cumulVal = false;
    if(i>=nodeNo) return false;
    for(int k=1; k<5; k++)
    {
        cumulVal |= Insert(4*i+k, x1, y1, x2, y2, ch);
    }
    if(cumulVal) return cumulVal;
    else return vect[i].Insert(x1, y1, x2, y2, ch);
};

void Tree::SetBBox(int i, double x1, double y1, double x2, double y2)
{
    if(i>=nodeNo) return;
    vect[i].SetBBox(x1, y1, x2, y2);
    double xs = (x1 + x2)/2.0;
    double ys = (y1 + y2)/2.0;

    SetBBox(4*i+1, x1, ys, xs, y2);
    SetBBox(4*i+2, xs, ys, x2, y2);
    SetBBox(4*i+3, x1, y1, xs, ys);
    SetBBox(4*i+4, xs, y1, x2, ys);
}

void Tree::Draw(int i)
{
    if (i>=nodeNo) return;
    vect[i].Draw();
    for(int k=1; k<5; k++) Draw(4*i+k);
}

int main(int argc, char* argv[])
{
    Tree t(3);
    t.SetBBox(0, 0, 0, 100, 100) ;
    t.Insert(0, 10, 10, 20, 20, 'A') ;
    t.Insert(0, 45, 45, 55, 55, 'B') ;
    t.Insert(0, 20, 20, 30, 30, 'C') ;
    t.Draw(0);
    return 0;
}

```

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

```
#include <stdio.h>

class A {
public:
    A(void);
    int x;
};

A::A (void)
{
    x = 7;
}

class B {
public:
    B (void);
    int y;
};

B::B (void)
{
    y = 1;
}

class C : public A, public B
{};

int A_function(A *a) {
    return a->x;
}

int B_function(B *b) {
    return b->y;
}

class D : public B, public A
{};

void main(void)
{
    C *c = new C();
    void *p = (void *) c;
    int x = A_function((A *) p);
    int y = B_function((B *) p);
    printf("%d\n%d\n", x, y);

    D *d = new D();
    p=(void *) d;
    y = A_function((A *) p);
    x = B_function((B *) p);
    printf("%d\n%d", x , y);
}
```

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

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

class Foo_Impl {
public:
    int x;
    void bar();
    void spring ();
};

void Foo_Impl::bar() { cout << "Struga " ; }
void Foo_Impl::spring() { cout << "Ohrid "; }

template<class T> class SmartPtr {
    T *pointee;
public:
    SmartPtr <T> (Foo_Impl *);
    T *operator-> () { return pointee; }
};

template <class T> SmartPtr <T> :: SmartPtr (Foo_Impl *f) {
    f->x = 10;
    cout << f -> x << " ";
}

typedef SmartPtr<Foo_Impl> Foo;

Foo make_Foo() {
    Foo_Impl *F = new Foo_Impl();
    return F;
}

void do_something(Foo f) {
    Foo_Impl *Dump = new Foo_Impl;
    SmartPtr <Foo_Impl> Dummy (Dump);
    Dummy.operator -> ()->spring ();
    f.operator ->() -> bar();
}

int main () {
    Foo_Impl *g = new Foo_Impl;
    Foo f (g);
    (f.operator->())->x;
    do_something (f);
    (f.operator->())->bar();

    return 0;
}
```

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

```
#include <iostream>

enum Currency {euro, dollar, dinar, denar};

using namespace std;

class EuropeanCountry {
protected:
    Currency c;
public:
    virtual void SetCur(Currency cur = euro) = 0;
    virtual Currency GetCur() {return euro;}
};

class WorldCountry{
protected:
    Currency c;
public:
    virtual void SetCur(Currency cur = dollar) = 0;
    virtual Currency GetCur() {return dollar;}
};

class Serbia : virtual public EuropeanCountry, public WorldCountry {
public:
    void SetCur(Currency cur = dinar)
        {WorldCountry::c = cur; EuropeanCountry::c = dinar;}
    Currency GetCur() {return WorldCountry::c;}
};

class Macedonia : public WorldCountry, public EuropeanCountry{
public:
    void SetCur(Currency cur = denar)
        {WorldCountry::c = cur; EuropeanCountry::c = euro;}
    Currency GetCur() {return WorldCountry::c;}
};

int main(int argc , char* argv[])
{
    Macedonia *m1;
    EuropeanCountry *m3, *s3;
    Serbia *s1;
    m1 = new Macedonia;
    s1 = new Serbia;
    m3 = new Macedonia;
    s3 = new Serbia;

    m1->SetCur();
    s1->SetCur();
    s3->SetCur();
    m3->SetCur();

    cout << "m1:" << m1->GetCur() << endl;
    cout << "m3:" << m3->GetCur() << endl;
    cout << "s1:" << s1->GetCur() << endl;
    cout << "s3:" << s3->GetCur() << endl;

    return 0;
}
```

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

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

class CKlasa {
private: static CKlasa * kopija;
public:
    CKlasa(int);
    CKlasa ();
    static CKlasa* Pokazivac();
    int Radnja();
};

CKlasa* CKlasa::kopija = NULL;
CKlasa :: CKlasa (int i ) {
    if (i > 2) CKlasa *p = new CKlasa (i - 2);
    cout << i << " ";
};

CKlasa :: CKlasa () { cout << "2 "; };

int CKlasa :: Radnja () {return ((int)kopija % 2) / 2;}

CKlasa* CKlasa :: Pokazivac () {
    if(kopija == NULL) kopija = new CKlasa (5);
    return kopija;
};

class CKlasa2 : CKlasa {
private: static CKlasa2 *kopija;
public:
    CKlasa2(int);
    static CKlasa2* Pokazivac();
    int Radnja();
};

CKlasa2* CKlasa2::kopija = NULL;

CKlasa2 :: CKlasa2 (int i) {
    if (i > 2) CKlasa *p = new CKlasa (i - 1);
    cout << i << " ";
};

int CKlasa2 :: Radnja () {
    CKlasa *p = new CKlasa (3);
    return ((int)kopija % 1) + 1;
};

CKlasa2* CKlasa2::Pokazivac () {
    if(kopija == NULL) kopija = (CKlasa2 *) new CKlasa(4);
    return kopija;
};

void main (void) {
    cout << CKlasa::Pokazivac()->Radnja() << " ";
    CKlasa2 *a = (CKlasa2 *) new CKlasa2 (10);
    CKlasa2 *b = (CKlasa2 *) new CKlasa (11);
    CKlasa2 *c = (CKlasa2 *) new CKlasa;
    cout << CKlasa2::Pokazivac()->Radnja() << " ";
    cout << a ->Radnja () << " ";
    cout << b ->Radnja () << " ";
    cout << c ->Radnja () << " ";
}
```



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

```
#include <iostream>

using namespace std;

class C
{
    int n;
public:
    C() { n = 0; }
    C(int n1) { n = n1; }
    virtual void print () { cout << "n = " << n << endl; }
    virtual void pr() { cout << "A\n"; }
    void nvpr() { cout << "C\n"; }
    int getn(int a) { return n ; }
};

class A : public C
{
    int m;
public:
    A() : C() { }
    A(int n1, int m1) : C(n1) { m = m1; }
    void print() {C::print(); cout << "m = " << m << endl; }
    virtual int print(int a) {int m2 = 1; C::print();
                                cout << "m = " << m2 << endl; return 0; }
    float getn(int a) { return (float)m; }
};

class B : public A
{
public:
    void pr() { cout << "B\n"; }
    void nvpr() { cout << "A\n"; };
};

int main () {
    C b;
    A c(2, 3);
    B a;
    b.print();
    c.print();
    c.print(0);
    cout << b.getn(0) << endl;
    cout << c.getn(0) << endl;
    b.pr();
    c.pr();
    a.pr();
    b.nvpr();
    c.nvpr();
    a.nvpr();
    a.print(0);
    C *bp = new B;
    bp -> pr();
    bp -> nvpr();
    delete bp;
    bp = new A;
    bp -> pr();
    bp -> nvpr();
    delete bp;
    return 0;
}
```

### **Task 8: What is the output of the following program? (ranije 5. zadatak)**

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

class Osoba2 {
public:
    Osoba2() {
        Tip = "Osoba2";
        cout << Tip;
        a = 10;
    }
    friend void Stampa(Osoba2& p);
    virtual int Go(const int, Osoba2 *);
    string Tip;
private:
    int a;
};

class Osobal : public Osoba2 {
public:
    Osobal()
    {
        Tip = "Osobal ";
        cout << Tip;
        a = 11;
    }
    friend void Stampa(Osoba2& p);
    virtual int Go (int, Osobal *);
    string Tip;
private:
    int a;
};

int Osoba2 :: Go (const int a, Osoba2 *m) {
    if (a > 0)
    {
        (*m).Go (a - 1, m) ;
        cout << a << " ";
    }
    return 0;
};

int Osobal :: Go (int p, Osobal *m) {
    if (a > 0)
    {
        Osoba2 :: Go(a - 1, m);
        cout << a << " ";
    }
    return 0;
};

void Stampa(Osoba2& p)
{ cout << p.Tip << endl; };

void main ()
{
    Osoba2 p;
    Osobal m;
    Stampa(p);
    Stampa(m);
    m.Go(7, (Osobal *) &p);
}
```

## Rešenja:

### Task1:

operator+ (A)  
operator- (g,A) ext  
operator+ (A)  
operator- (g,A) ext  
6001

### Task2:

0 0 100 100 B  
0 0 50 50 C  
0 0 25 25 A

### Task 3:

7  
7  
1  
1

### Task 4:

10 10 Ohrid Struga Struga

### Task 5:

m1 : 3  
m3 : 0  
s1 : 2  
s3 : 0

### Task 6:

1 3 5 0 2 1 3 5 7 9 10 1 3 5 7 9 11 2 2 4 1 3 1 1 3 1 1 3 1  
1 3 1

**Task 7:**

n = 0

n = 2

m = 3

n = 2

m = 1

0

3

A

A

B

C

C

A

n = 0

m = 1

B

C

A

C

**Task 8:**

Osoba2Osoba2Osoba1Osoba2

Osoba2

1 2 3 4 5 6 7 8 9 10 11