Elektrijada 2005 **Kopaonik**

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;

int mData = 20;

class UPInt {
public:
      UPInt& operator++() {cout << mData-- <<endl; UPInt* p = new UPInt; return *p;}
      const UPInt operator++(int) {cout << mData++ << endl; return *this;}
      UPInt& operator--() {cout << ++mData << endl; UPInt* p = new UPInt; return *p;}
      const UPInt operator--(int) {cout << mData-- << endl; return *this;}
      UPInt& operator+=(int);
};

void main() {
      UPInt i;
      UPInt j;
      ++i;
      j++;
      --j;
      i++;
      --i;
      ++j;
      j--;
      i--;
}
```

Task2: What is the output of the following program?

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

class A {
public:
      int a;
      A() { a = 0; }
      A(int a1) { a = a1; }
      void print() { cout << "A::a = " << a << endl; }
};

class B : virtual public A {
      int b;
public:
      B() : A() { b = 0 ; }
      B(int a1, int b1) : A(a1) { b = b1; }
      void print() { cout << "B::"; A::print(); cout << "B::b = " << b << endl; }
};

class C : virtual public A {
      int c;
public:
      C() : A() { c = 0; }
      C(int a1, int c1) : A(a1) { c = c1; }
      void print() { cout << "C::"; A::print(); cout << "C::c = " << c << endl; }
};

class D : public B, public C {
      int d;
public:
      D() : B(), C() { d = 0 ; }
      D(int a1, int a2, int b1, int c1, int dl) : B(a1, b1), C(a2, c1) { d = dl; }
      void print() { B::print(); C::print(); cout << "D::d = " << d << endl; }
};

class E : public A {
      int e;
public:
      E() : A() { e = 0; }
      E(int a1, int e1) : A(a1) { e = e1; }
      void print() { cout << "E::"; A::print(); cout << "E::e = " << e << endl; }
};

class F : public B, public C, public E {
      int f;
public:
      F() : B(), C(), E() { f = 0; }
      F(int a1, int a2, int a3, int b1, int c1, int e1, int f1) : B(a1, b1), C(a2,
c1), E(a3, e1) { f = f1; }
      void print() { B::print(); C::print(); E::print(); cout<< "F::f = " << f <<
endl; }
};

void main () {
      D x(1, 2, 3, 4, 5);
      x.print(); x.a = 7; x.B::a = 8;
      x.C::a = 9; x.print();
      F y(1, 2, 3, 4, 5, 6, 7);
      y.print(); y.B::a = 1;
      y.E::a = 2; y.print();
}
```

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

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

char outputString[30];
static int i = 0;

class Opel;

class Car {
public:
      Car(){outputString[i++]='1';}
      ~Car(){outputString[i++]='A';}
};

const Opel& returnCar(const Opel& p) {return p;}

class Opel : public Car {
public:
      Opel() {outputString[i++]='2';}
      ~Opel() {outputString[i++]='B';cout << "Opel " << endl;}
};

Opel returnOpel(Opel s){ return s;}

class Astra : public Opel {
public:
      Astra(){outputString[i++]='3';}
      ~Astra(){outputString[i++]='C';cout << "Astra " << endl;}
};

Opel returnAstra(Opel s) { return s;}

class Vectra : public Astra {
public:
      Vectra(){outputString[i++]='4';}
      ~Vectra(){outputString[i++]='D';cout << "Vectra " << endl;}
};

Car returnVectra(Car d) { return d;}

int main(int argc, char* argv[])
{
      Vectra laza;
      Vectra* pLaza;
      pLaza = &laza;
      returnVectra(returnAstra(returnOpel(returnCar(*pLaza))));
      cout << outputString << i << endl;
      return 0;
}
```

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

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

enum ShapeColor {RED, GREEN, BLUE};

class ShapeA {
public:
      virtual void draw(ShapeColor color = RED) = 0;
};

class ShapeB : public ShapeA {
public:
      virtual void draw(ShapeColor color = GREEN) {cout << "ShapeB-draw::" << color
<< endl;}
};

class ShapeC : public ShapeB {
public:
      virtual void draw(ShapeColor color = BLUE) {cout << "ShapeC-draw::" << color <<
endl;}
};

int main(int argc, char argv[])
{
      ShapeA *ps2 = new ShapeC;
      ShapeA *ps3 = new ShapeB;
      ps2->draw(RED);
      ps3->draw(BLUE);
      ps3->draw();
      ps2->draw();

      return 0;
}
```

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

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

class Base {
      int x;
      virtual int f( int );
public:
      Base() {}
      Base( int a ) : x(a) {cout << "Base" << endl;}
      double f( double );
};

int Base::f( int a ) { return a+x;}
double Base::f( double a ) { return a*x + f(5);}

class Derived : virtual public Base{
      int x;
      int f( int );
public:
      Derived ( int a, int b ) : Base(a), x(b) {cout << "Derived" << endl;}
      double f( double );
};

int Derived::f( int a ) { return a-x;}
double Derived::f( double a ) { return a/x+f(5);}

class NewDerivedA : virtual public Base{
      int x;
public: NewDerivedA ( int a, int b ) : Base(a), x(b) {cout << "NewDerivedA" << endl;}
};

class NewDerivedB : virtual public Base{
      int x;
public: NewDerivedB ( int a, int b ) : Base(a), x(b) {cout << "NewDerivedS" << endl;}
};

class All : public NewDerivedA, public NewDerivedB {
public:
      All(int a, int b) : NewDerivedA(a,b) , NewDerivedB(b,a){ cout<< "All" << endl;}
      double g(double a, double b , double f) { return a-2*b-f;}
};

void main() {
      Base b(4);
      Derived d(4,8);
      All pa(4, 8);
      Base* pb = new Derived(4,8);
      cout<< pa.g(b.f( 1.2 ) ,d.f( 1.2 ), pb->f( 1.2 )) << endl;
}
```

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

```
#include <iostream>

#include <string>

using namespace std;

class CSTRING {
protected:
      char *niz;
      int duz;
public:
      CSTRING() {};
      void Postavi(char* s1, int n);
      ~CSTRING();
      void operator--();
      int tezina (void);
};

int CSTRING::tezina(void) { return duz * 3;}

void CSTRING::Postavi(char *sl, int n) {
      int i;
      duz=n;
      niz=new char[duz];
      for(i=0;i<duz;i++)
            niz[i]=sl[i];
}

CSTRING::~CSTRING() { delete[] niz; duz=0; }

void CSTRING::operator --() {
      int i, j;
      char* pomniz=new char[duz];
      for(i=0;i<duz;i++)
            pomniz[i]=niz[i];
      for(i=0;i<duz;i++)
            if(pomniz[i]=='A' || pomniz[i]=='a' || pomniz[i]=='E' || pomniz[i]=='e'
|| pomniz[i]=='I' || pomniz[i]=='i' || pomniz[i]=='O' || pomniz[i]=='o' ||
pomniz[i]=='U' || pomniz[i]=='u') {
                        if (i!=duz-1)
                              for(j=i+1;j<duz;j++)
                                    pomniz[j-1]=pomniz[j];
                        duz--;
            }
      for(i=0;i<duz;i++)
            niz[i]=pomniz[i];
      delete[] pomniz;
      return;
}

class CNIZ {
      int *niz;
      int n;
public:
      CNIZ(int n1);
      ~CNIZ();
      void Sort();
      void Upis();
      void Stampanje();
};
```

```
CNIZ::CNIZ(int n1) { n=n1; niz=new int[n]; }

CNIZ::~CNIZ() { delete[] niz; }

void CNIZ::Sort() {
      int i,j;
      for (i=0;i<n-1;i++)
            for(j=i;j<n;j++)
                  if (niz[i]>=niz[j]) {
                        niz[i]=niz[i]+niz[j];
                        niz[j]=niz[i]-niz[j];
                        niz[i]=niz[i]-niz[j];
                  }
      return;
}

void CNIZ::Upis() {
      CSTRING *stringovi;
      int i;
      stringovi = new CSTRING[3];
      char *sadrzaj1 = new char[8];
      char *sadrzaj2 = new char[4];
      char *sadrzaj3 = new char[10];

      sadrzaj1 = "KOPAONIK";
      sadrzaj2 = "2005";
      sadrzaj3 = "ELEKTRIJADA";

      stringovi[0].Postavi(sadrzaj1,8);
      stringovi[1].Postavi(sadrzaj2,4);
      stringovi[2].Postavi(sadrzaj3,5);
      for(i=0;i<3;i++) --stringovi[i];

      for(i=0;i<n;i++)
            niz[i]=2*stringovi[i%3].tezina();
      return;
}

void CNIZ::Stampanje() {
      int i;
      for(i=0;i<n;i++)
            cout<<niz[i]<<" ";
      cout<<"\n";
      return;
}

void main() {
      int br1=7;
      CNIZ niz1(br1);
      niz1.Upis();
      niz1.Sort();
      niz1.Stampanje();
}
```

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

```
#include <iostream>

using namespace std;

int n = 3, h0 = 512, equalCoo = 0;
static int mX,mY,mH;

class DrawA;
class DrawB;
class DrawC;
class DrawD;

void equal(){if(mX == 128) {equalCoo++;cout << mY << endl;}}

class DrawA {
public:
      void drawA(int);
};

class DrawB {
public:
      void drawB(int);
};

class DrawC {
public:
      void drawC(int);
};

class DrawD {
public:
      void drawD(int);
};


void DrawA::drawA(int i) {
      if(i==0) return;
      drawA(i-1); mX+=mH; mY-=mH; equal();
      DrawB b; b.drawB(i-1); mX+=2*mH; equal();
      DrawD d; d.drawD(i-1); mX+=mH; mY+=mH; equal();
      drawA(i-1); equal();
};


void DrawB::drawB(int i) {
      if(i==0) return;
      drawB(i-1); mX-=mH; mY-=mH; equal();
      DrawC c; c.drawC(i-1); mX-=2*mH; equal();
      DrawA a; a.drawA(i-1); mX+=mH; mY-=mH; equal();
      drawB(i-1); equal();
}


void DrawC::drawC(int i) {
      if(i==0) return;
      drawC(i-1); mX-=mH; mY+=mH; equal();
      DrawD d; d.drawD(i-1); mX-=2*mH; equal();
      DrawB b; b.drawB(i-1); mX-=mH; mY-=mH; equal();
      drawC(i-1); equal();
};
```

```
void DrawD::drawD(int i) {
      if(i==0) return;
      drawD(i-1); mX+=mH; mY+=mH; equal();
      DrawA a; a.drawA(i-1); mX+=2*mH; equal();
      DrawC c; c.drawC(i-1); mX-=mH; mY+=mH; equal();
      drawD(i-1); equal();
};


int main(int arqc, char* argv[])
{
      int mH = h0 / 4, x0 = 2*mH, y0 = 3*mH, i = 0;
      int brprolaza = 0;
      while(i!=n){
            i++;
            x0-=mH;
            mH=mH/2;
            y0+=mH;
            mX=x0;
            mY=y0;
            DrawA A;
            A.drawA(i);
            mX+=mH;
            mY-=mH;
            DrawB B;
            B.drawB(i);
            mX-=mH;
            mY-=mH;
            DrawC C;
            C.drawC(i);
            mX-=mH;
            mY+=mH;
            DrawD D;
            D.drawD(i);
            mX+=mH;
            mY+=mH;
      }
      cout << equalCoo << endl;
      return 0;
}
```

Rešenja:

Task1:

```
20
19
21
21
23
23
22
21
```

Task2:

```
B::A::a = 0
B::b = 3
C::A::a = 0
C::c = 4
D::d = 5
B::A::a = 9
B::b = 3
C::A::a = 9
C::c = 4
D::d = 5
B::A::a = 0
B::b = 4
C::A::a = 0
C::c = 5
E::A::a = 3
E::e = 6
F::f = 7
B::A::a = 1
B::b = 4
C::A::a = 1
C::c = 5
E::A::a = 2
E::e = 6
F::f = 7
```

Task 3:

```
Opel
Opel
Opel
Opel
1234BABAAAABABA15
Vectra
Astra
Opel
```

Task 4:

```
ShapeC-draw::0
ShapeB-draw::2
ShapeB-draw::0
ShapeC-draw::0
```

Task 5:

```
Base
Base
Derived
NewDerivedA
NewDerivedB
All
Base
Derived
17.7
```

Task 6:

```
0 0 0 0 0 0 30
```

Task 7:

```
448
448
448
448
320
320
320
320
8
```