

Lösningsförslag till datorövning 2, Objektorienterad programmering

```
// OOP2ABMA.cpp

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

void swap(float *xp, float *yp)
{
    float z;

    z = *xp;
    *xp = *yp;
    *yp = z;
}

void swap(float &xr, float &yr)
{
    float z;

    z = xr;
    xr = yr;
    yr = z;
}

void swap(char *s1, char *s2)
{
    char temp[100];

    strcpy(temp, s1);
    strcpy(s1, s2);
    strcpy(s2, temp);
}

void main()
{
    float x, y;
    char namn1[100], namn2[100];

    cout << "Ge två tal x y : " ;
    cin >> x >> y;
    swap(&x, &y);
    cout << "Talen x y : " << x << ' ' << y << endl;
    swap(x, y);
    cout << "Talen x y : " << x << ' ' << y << endl;
    cin.get();
    cout << "Ge sträng1 : ";
    cin.getline(namn1, 100);
    cout << "Ge sträng2 : ";
    cin.getline(namn2, 100);
    swap(namn1, namn2);
    cout << namn1 << ' ' << namn2 << endl;
}

//OOP2CMA.cpp

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

```

float medel(int tv[], int nr = 1000)
{
    float summa = 0.0;

    for (int i = 0; i < nr; i++)
    {
        summa += tv[i];
    }
    return summa / nr;
}

void main()
{
    int antal, *tvek;

    srand((unsigned)time(NULL));
    cout << "Ge antal kast : ";
    cin >> antal;
    tvek = new int[antal];
    for (int i = 0; i < antal; i++)
    {
        tvek[i] = rand() % 6 + 1;
    }
    cout << "Medel : " << medel(tvek, antal) << endl;
    delete [] tvek;
    getch();
}

```

//OOP2DMA.cpp

```

#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;

float medel(int tv[], int nr = 1000)
{
    float summa = 0.0;

    for (int i = 0; i < nr; i++)
    {
        summa += tv[i];
    }
    return summa /nr;
}

void main(int argc, char *argv[])
{
    int antal, *tvek;

    if (argc == 1)
    {
        antal = 1000;
    }
    else if (argc == 2)
    {
        antal = atoi(argv[1]);
    }
    else
    {

```

```

        cerr << "För många parametrar till main!" << endl;
        exit(1);
    }
    srand((unsigned)time(NULL));
    tvek = new int[antal];
    for (int i = 0; i < antal; i++)
    {
        tvek[i] = rand() % 6 + 1;
    }
    cout << "Medel : " << medel(tvek, antal) << endl;
    delete [] tvek;
    getch();
}

```

//OOP2E.cpp

```

#include "oop2e.h"
#include <iostream.h>

```

```

void VEKTOR::skapa(int s)
{
    storlek = s;
    vp = new float[s];
}

```

```

void VEKTOR::las()
{
    for (int i = 0; i < storlek; i++)
    {
        cout << "vp[" << i << "] =";
        cin >> vp[i];
    }
}

```

```

void VEKTOR::skriv()
{
    for (int i = 0; i < storlek; i++)
        cout << vp[i] << endl;
}

```

```

void VEKTOR::tabort()
{
    storlek = 0;
    delete [] vp;
    vp = NULL;
}

```

OOP2EMA.cpp

```

#include "oop2e.h"

```

```

void main()
{
    VEKTOR v;

    v.skapa(3);
    v.las();
    v.skriv();
    v.tabort();
}

```

```

//OOP2F.cpp

void MATRIS::skapa(int r, int k)
{
    rad = r;
    kol = k;
    mat = new float*[rad];
    for (int i = 0; i < kol; i++)
        mat[i] = new float[k];
}

void MATRIS::tabort()
{
    for (int i = 0; i < rad; i++)
        delete [] mat[i];
    delete [] mat;
    mat = NULL;
    rad = kol = 0;
}

void MATRIS::las()
{
    for (int i = 0; i < rad; i++)
        for (int j = 0; j < kol; j++)
            {
                cout << "Matris[" << i << "][" << j << "] = ";
                cin >> mat[i][j];
            }
}

void MATRIS::skriv()
{
    for (int i = 0; i < rad; i++)
        {
            for (int j = 0; j < kol; j++)
                {
                    cout.width(5);
                    cout << mat[i][j];
                }
            cout << endl;
        }
}

OOPP2FMA.cpp

#include "oop2f.h"

void main()
{
    MATRIS m;

    m.skapa(2, 3);
    m.las();
    m.skriv();
    m.tabort();
}

```