

## Lösningsförslag till datorövning 1, Objektorienterad programmering

```
// OOP1AMA.cpp

#include <iostream>
#include <conio.h>
using namespace std;

void main()
{
    char namn[20];

    cout << "Ge ditt namn : ";
    cin.getline(namn, 20);
    gotoxy(30, 12);
    cout << namn;
    getch();
}

// OOP1BMA.cpp

#include <iostream>
using namespace std;

int sgd(int m, int n)
{
    int rest;

    while (n!= 0)
    {
        rest = m % n;
        m = n;
        n = rest;
    }
    return m;
}

void main()
{
    int t, n, del;
    char ch;

    cout << "Ge ett bråk : ";
    cin >> t >> ch >> n;
    del = sgd(t, n);
    cout << "Bråk : " << t << '/' << n << endl;
    cout << "Förkortat : " << t/del << '/' << n/del << endl;
}

// OOP1C.cpp

#include "oop1c.h"
#include <iostream>
using namespace std;

void CAR::las()
```

```

{
    cout << "Ge regnr : ";
    cin >> regnr;
    cout << "Ge effekt : ";
    cin >> power;
    cout << "Ge antal cylindrar : ";
    cin >> nrcyl;
}

void CAR::skriv()
{
    cout << "Regnr : " << regnr;
    cout << "Effekt : " << power;
    cout << "Antal cylindrar : " << nrcyl;
}

bool CAR::is_less_power(CAR c)
{
    return power < c.power;
}

```

//OOP1CMA.cpp

```
#include "oop1c.h"
```

```

void main()
{
    CAR ca, cb;

    ca.las();
    cb.las();

    if (ca.is_less_power(cb))
    {
        ca.skriv();
        cb.skriv();
    }
    else
    {
        cb.skriv();
        ca.skriv();
    }
}

```

//OOP1D.cpp

```

#include "oop1d.h"
#include <iostream>
using namespace std;

```

```

int sgd(int m, int n)
{
    int rest;

    while (n!= 0)
    {
        rest = m % n;
        m = n;
        n = rest;
    }
}

```

```

    }
    return m;
}

void RTAL::las()
{
    char ch;

    cout << "Ge ett bråk : ";
    cin >> t >> ch >> n;
}

void RTAL::skriv()
{
    cout << "Bråket : " << t << '/' << n << endl;
}

void RTAL::forkorta()
{
    int del = sgd(t, n);

    t /= del;
    n /= del;
}

RTAL RTAL::add(RTAL rt)
{
    RTAL res;

    res.t = t*rt.n + n*rt.t;
    res.n = n*rt.n;
    res.forkorta();
    return res;
}

RTAL RTAL::sub(RTAL rt)
{
    RTAL res;

    res.t = t*rt.n - n*rt.t;
    res.n = n*rt.n;
    res.forkorta();
    return res;
}

RTAL RTAL::mul(RTAL rt)
{
    RTAL res;

    res.t = t*rt.t;
    res.n = n*rt.n;
    res.forkorta();
    return res;
}

RTAL RTAL::div(RTAL rt)
{
    RTAL res;

```

```
    res.t = t*rt.n;  
    res.n = n*rt.t;  
    res.forkorta();  
    return res;  
}
```

```
// OOP1DMA.cpp
```

```
#include "oop1d.h"
```

```
void main()  
{  
    RTAL a, b, c, r;  
  
    a.las();  
    b.las();  
    c.las();  
    r = a.add(b.mul(c));  
    r.skriv();  
}
```