



## Dag 5

Pedher Johansson (pedher@cs.umu.se)  
Thomas Johansson (thomasj@cs.umu.se)  
Mattias Åsander (mattiasa@cs.umu.se)

23 maj 2014 Kurs i C++

1



## Dag 5

- Överlagring av funktioner
- Referenser

23 maj 2014 Kurs i C++

2



## Överlagring

- I C++ kan man skapa flera metoder med samma namn och returvärde, men med olika parametrar
- Vilken metod som anropas bestäms vid anropet

23 maj 2014 Kurs i C++

3



```
class Timestamp
{
private:
    int hours;
    int minutes;
public:
    ...
    void step(int h, int m);
    void step(int m);
};

Timestamp::step(int h, int m)
{
    int totm = (hours+h) * 60 +
               (minutes + m);

    hours = (totm / 60) % 24;
    minutes = totm % 60;
}

Timestamp::step(int m)
{
    step(0, m);
}
...
```

```
Timestamp t(23, 20);
t.step(2, 0);
t.step(120);
```

23 maj 2014 Kurs i C++

4



## Defaultkonstruktör

- Alla klasser bör implementera en default-konstruktör som inte kräver argument.
  - Används t.ex. när man vill skapa vektorer av objekt

```
Timestamp vec[10];
```

23 maj 2014 Kurs i C++

5



```
class Timestamp
{
private:
    int hours;
    int minutes;
public:
    Timestamp();
    Timestamp(int h, int m);
    ...
};

Timestamp::Timestamp()
{
    hours = 0;
    minutes = 0;
}

Timestamp::Timestamp(int h, int m)
{
    hours = h;
    minutes = m;
}
...
```

```
Timestamp t1;
Timestamp t2(23, 12);
```

23 maj 2014 Kurs i C++

6



## Referenser

- C++ har två(!) sätt att referera till ett annan variabels adress i minnet

- Pekare
- Referens

```
int a = 3;
int* pa = &a;

*pa = 5;
cout << a << endl; // 5
cout << pa << endl; // adress
```

```
int a = 3;
int& ra = a;

ra = 5;
cout << a << endl; // 5
cout << ra << endl; // 5
```

23 maj 2014

Kurs i C++

7



## Referenser

- Referenser används främst i metoder där man vill undvika kopiering av objekt men slippa pekarnotation
- Ett exempel är en copy-konstruktor

23 maj 2014

Kurs i C++

8



```
class Timestamp
{
private:
    int hours;
    int minutes;
public:
    Timestamp();
    Timestamp(int h, int m);
    Timestamp(Timestamp& t);
    ...
};

Timestamp::Timestamp()
{
    hours = 0;
    minutes = 0;
}

Timestamp::Timestamp(int h, int m)
{
    hours = h;
    minutes = m;
}

Timestamp::Timestamp(Timestamp& t)
{
    hours = t.minutes;
    minutes = t.hours;
}
```

```
Timestamp t1(23, 20);
Timestamp t2(t1);
```

23 maj 2014

Kurs i C++

9



## Referenser

- Referenser kan även användas till att ändra ett arguments globala värde
  - Sällan nödvändigt och **ofta olämpligt**

```
void foo(int& a)
{
    a = 5;
}

int a = 3;
foo(a);
cout << a << endl; // 5
```

23 maj 2014

Kurs i C++

10



## Referenser

- Referenser kan inte
  - byta refererad variabel
  - sättas till null
  - lämnas utan värde

```
int a = 3;
int& ra = a; // Tillåtet
```

```
int& ra; // Ej tillåtet
```

```
int a;
int& ra = a; // Tillåtet !!!
```

23 maj 2014

Kurs i C++

11