# Tillämpad programmering ID1218

C++ Introduktion

Johan Montelius

# Hello World

```
#include <iostream>

using namespace std;

// Hello

int main() {
   cout << "Hello world." << endl;
   return 0;
}
```

# #include

```
#include <iostream>

using namespace std;

// Hello

int main() {
   cout << "Hello world." << endl;
   return 0;
}
```

```
g++ -E -o main.pp main.cc
```

# Preprocessor

```
#include <iostream>

#include “foo.h”

 :
 :
         #ifdef DEBUG
             :
             :

         #else
             :

         #endif
```

# int main ()

```
#include <iostream>

using namespace std;

// Hello

int main() {
    cout << "Hello world." << endl;
    return 0;
}
```

# int main

```
int main(int argc, char *argv[]) {
        :
        :

}
```

# namespaces

```
#include <iostream>

using namespace std;

// Hello

int main() {
    cout << "Hello world." << endl;
    return 0;
}
```

# std::cout

```
#include <iostream>

int main() {
    std::cout << "Hi" << std::endl;
    return 0;
}
```

# std::cout

```
#include <iostream>

int main() {
    std::cout << "Hi" << std::endl;
    return 0;
}
```

# kommentarer

```
#include <iostream>

using namespace std;

// Hello

int main() {
   cout << "Hello world." << endl;
   return 0;
}
```

# kommentarer

```
int x;  // x är någonting
  :



           /* En kommentar
              som sträcker sig över
              flera rader
           */
             :
```

# datatyper

- bool
  - boolska värden : true, false
- char, ...
  - tecken : 'a', 'b', ...
- (signed) short, int, long
  - heltal : -20, -024, 0x14
- unsigned short, int, long
  - positiva heltal: 20, 024, 0x14
- (signed/unsigned) float double
  - flyttal: 3.14, 2.3456e4

# variabler

- deklareras
  - int x;
- initialiseras
  - int x = 5 ;
  - int x(5);
- tilldelas
  - x = 45;

# räckvidd variabler

```
int i = 25;

int main() {
     int i = 5;

     for(int i = 0; i < 2, i++) {
        std:cout << i << std:endl;
     }

     std:cout << i << std:endl;

     return 0;
}
```

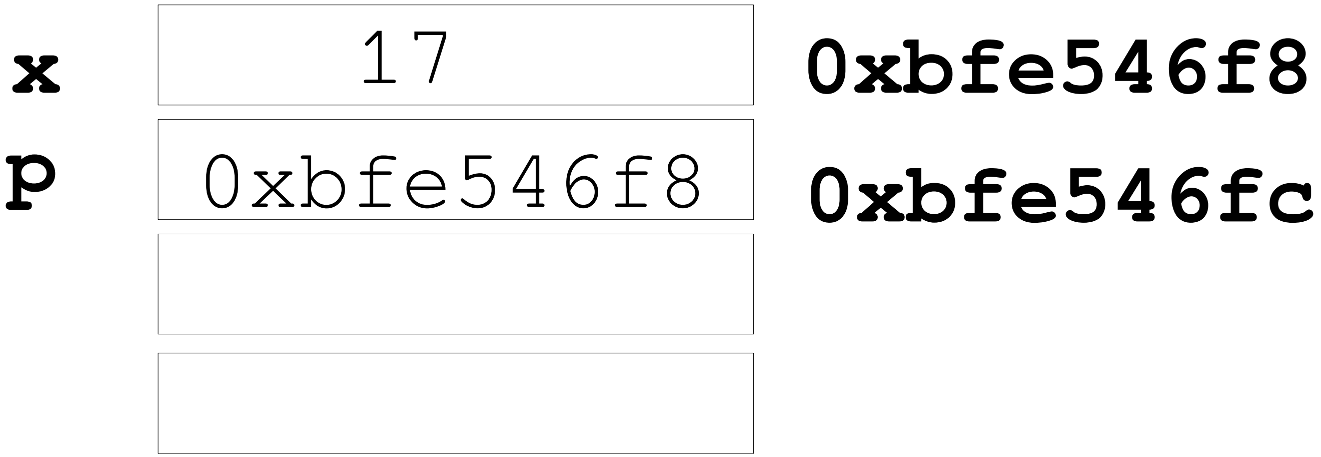
# referenser (alias)

```
int main() {
        int i = 5;

        int &x = i;

        cout << “x = “ << x << endl;

        return 0;

}
```

# fel

```
int main() {
      int &x = 10;
      cout << “x = “ << x << endl;
      return 0;
}
```

# pekare



| | | |
|---|---|---|
| x | 17 | 0xbfe546f8 |
| p | 0xbfe546f8 | 0xbfe546fc |
| | | |
| | | |

```
int main() {
    int x = 17;
    int *p;
    p = &x;
}
```

# förvirring

```
int i = 42;

int &r = i;
int *p;

p = &i;
*p = 17;

int &rp = *p;
rp = 15;
```

# källa till fel

```
int i = 42;

int* p;
int *p;

int i, j, k;

int* p, q;

int *p, *q;
```

# pekare till pekare

```
int i = 42;

int *p = &i;
int **q = &p;

cout << i << endl;

cout << *p << endl;

cout << **q << end;
```

# const

```
int main() {
    const float i = 3.14;

        :

}
```

# typedef

```
#ifdef OLD_MACHINE

    typedef unsigned long uint32;

#else

    typedef unsigned int uint32;

#endif
```

# typedef

```
typedef int weight;
typedef int price;
typedef int cost;

const price ap = 42;

price buy(weight apples) {
    return (apples * ap)
}
```

# struct

```
struct Fruit {
    std::string name;
    int price;
};

Fruit apple, banana;

int cost(Fruit f, int amount) {
    f.price * amount;
}
```

# if, for, while

```
if(i = 10) {
    do_this();
} else {
    do_that();
}
```

# const

```
int main() {
    const int i = 42;
    const int j = foo(42);
              :
}
```

# Nästa föreläsning

- Funktioner
  - hur ges argument
  - hur returneras värden
- Klasser
  - definitioner
  - hur skapas objekt