Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Introduction to C

Welcome to R'lyeh

Torje Hoås Digernes

Programvareverkstedet - PVV

Trondheim 2017

Outline

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C
Why C
IO
Minnehåndtering
Sortering

1. Comparison between Languages
   - Python and C
   - Java and C
2. Why C
   - Is there any pros at all?
   - Funksjonskall
3. IO
   - IO functions
   - `printf()` example
   - `scanf()` example
4. Minnehåndtering
   - pekere
   - struct
5. Sortering
   - Funksjonspekere

Python and C
Optional Subtitle

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management

Python and C
Optional Subtitle

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management
- Archaic module system (glorified copy paste)

Python and C
Optional Subtitle

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management
- Archaic module system (glorified copy paste)
- limited standard library

Python and C
Optional Subtitle

Introduction to C
Torje Hoås Digernes
Comparison between Languages
Python and C
Java and C
Why C
IO
Minnehåndtering
Sortering

- Manual memory management
- Archaic module system (glorified copy paste)
- limited standard library
- No builtin array

Outline

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C
Why C
IO
Minnehåndtering
Sortering

1. Comparison between Languages
   - Python and C
   - Java and C
2. Why C
   - Is there any pros at all?
   - Funksjonskall
3. IO
   - IO functions
   - `printf()` example
   - `scanf()` example
4. Minnehåndtering
   - pekere
   - struct
5. Sortering
   - Funksjonspekere

Java and C

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management

Java and C

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Python and C

Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management
- Archaic module system (glorified copy paste)

Java and C

Introduction to C

Torje Hoås Digernes

Comparison between Languages
Python and C
Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management
- Archaic module system (glorified copy paste)
- limited standard library

Java and C

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Python and C

Java and C

Why C

IO

Minnehåndtering

Sortering

- Manual memory management
- Archaic module system (glorified copy paste)
- limited standard library
- No builtin array

Outline

1 Comparison between Languages
- Python and C
- Java and C

2 Why C
- Is there any pros at all?
- Funksjonskall

3 IO
- IO functions
- `printf()` example
- `scanf()` example

4 Minnehåndtering
- pekere
- struct

5 Sortering
- Funksjonspekere

Is there any pros at all?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- Blazing speed

Is there any pros at all?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- Blazing speed (produces SEGFAULTS faster than you can imagine)

Is there any pros at all?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- Blazing speed (produces SEGFAULTS faster than you can imagine)
- Actual threading

Is there any pros at all?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- Blazing speed (produces SEGFAULTS faster than you can imagine)
- Actual threading (what is Global Interpreter Lock, snakey?)

Is there any pros at all?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- Blazing speed (produces SEGFAULTS faster than you can imagine)
- Actual threading (what is Global Interpreter Lock, snakey?)
- bare bones,

Is there any pros at all?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- Blazing speed (produces SEGFAULTS faster than you can imagine)
- Actual threading (what is Global Interpreter Lock, snakey?)
- bare bones, (low startup and time and low resource usage. Looking at you Java)

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

```c
#include <stdio.h>

int main(){
    printf("%s\n", "Hello, World! \n");
    return 0;
}
```

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

```c
#include <stdio.h>

int main(){
    printf("%s\n", "Hello, World! \n");
    return 0;
}
```

More pros

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

- So close to hardware you can actually feel the current passing through the processor
- Perfect for reinventing the entire wheel, spokes, hub, and rim

Outline

1. Comparison between Languages
   - Python and C
   - Java and C
2. Why C
   - Is there any pros at all?
   - Funksjonskall
3. IO
   - IO functions
   - `printf()` example
   - `scanf()` example
4. Minnehåndtering
   - pekere
   - struct
5. Sortering
   - Funksjonspekere

Call by value

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

Is there any pros at all?

Funksjonskall

IO

Minnehåndtering

Sortering

alt er call by value, selv pekere

Outline

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

IO functions

printf() example

scanf() example

Minnehåndtering

Sortering

1 Comparison between Languages
- Python and C
- Java and C

2 Why C
- Is there any pros at all?
- Funksjonskall

3 IO
- IO functions
- `printf()` example
- `scanf()` example

4 Minnehåndtering
- pekere
- struct

5 Sortering
- Funksjonspekere

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

IO functions

printf() example

scanf() example

Minnehåndtering

Sortering

printf(const char * format, ...)

Skriver ut ... i henhld til format

scanf(const char * format, ...)

Leser input i henhold til format og lagrer til ...

Outline

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

IO functions

printf() example

scanf() example

Minnehåndtering

Sortering

1 Comparison between Languages
- Python and C
- Java and C

2 Why C
- Is there any pros at all?
- Funksjonskall

3 IO
- IO functions
- printf() example
- scanf() example

4 Minnehåndtering
- pekere
- struct

5 Sortering
- Funksjonspekere

printf() example

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

IO functions

printf() example

scanf() example

Minnehåndtering

Sortering

```c
#include <stdio.h>
#define M_PI  3.14159;
double sin(double x);
double cos(double x);

int main(){
    double pi = M_PI;
    printf("sin(pi/4) = %f  -- cos(pi/4) = %f\n", sin
    return 0;
}
```

printf() example

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

IO functions

printf() example

scanf() example

Minnehåndtering

Sortering

```c
#include <stdio.h>
#include <math.h>

int main(){
    double pi = M_PI;
    printf("sin(pi/4) = %f  -- cos(pi/4) = %f \n", si
    return 0;
}
```

Outline

1 Comparison between Languages
- Python and C
- Java and C

2 Why C
- Is there any pros at all?
- Funksjonskall

3 IO
- IO functions
- printf() example
- scanf() example

4 Minnehåndtering
- pekere
- struct

5 Sortering
- Funksjonspekere

scanf() example

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

IO functions

printf() example

scanf() example

Minnehåndtering

Sortering

```c
#include <stdio.h>
#include <math.h>

int main(){
    double pi = M_PI;
    printf("sin(pi/4) = %f  -- cos(pi/4) = %f \n", si
    return 0;
}
```

malloc

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

```
void *malloc(size_t bytes)
```

Ber operativsystemet om nytt minne og returnerer en peker til det

Outline

1. Comparison between Languages
   - Python and C
   - Java and C
2. Why C
   - Is there any pros at all?
   - Funksjonskall
3. IO
   - IO functions
   - `printf()` example
   - `scanf()` example
4. Minnehåndtering
   - pekere
   - struct
5. Sortering
   - Funksjonspekere

pekere

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

- endrer du på pekeren selv endrer du bare hvor du peker
- derefererer du først endrer du på det den peker på
- to måter å dereferere på: [], *

malloc eksempel

Introduction to C
Torje Hoås Digernes
Comparison between Languages
Why C
IO
Minnehåndtering
pekere
struct
Sortering

```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    int length;
    printf("length: ");
    scanf("%i\n",&length );
    int * a = malloc(length*sizeof(int));
    for (size_t i = 0; i < length; i++) {
        scanf("%i ", a+i );
    }for (size_t i = 0; i < length; i++) {
        printf("%i ", a[i] );
    }
    return 0;
}
```

Outline

1. Comparison between Languages
   - Python and C
   - Java and C
2. Why C
   - Is there any pros at all?
   - Funksjonskall
3. IO
   - IO functions
   - `printf()` example
   - `scanf()` example
4. Minnehåndtering
   - pekere
   - struct
5. Sortering
   - Funksjonspekere

struct

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

egendefinerte datatyper som inneholder variabler.

struct square

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

input examples struct square med utskrift og setting

Mer abstract struct: tabell med lengde

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

- Pekere vet ikke lengde

Mer abstract struct: tabell med lengde

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

- Pekere vet ikke lengde
- vi kan kombinere pekere og lengde i en struct

Mer abstract struct: tabell med lengde

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

- Pekere vet ikke lengde
- vi kan kombinere pekere og lengde i en struct og det kan være lurt å ta med brukt mengde

struct Array

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

struct Array

input example Array

ekstra funksjoner til Array

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

- push
- resize
- pop

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

push,pop, resize example

filter

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

plukk ut positive tall fra en liste

filterfunksjon

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

inputex print all numbers larger than 0

fitlerfunksjon

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

pekere

struct

Sortering

inputex push all numbers larger than 0 to Array

qsort

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

```
void qsort(void *base, size_t nel, size_t width,
int (*compar)(const void *, const void *));
```

qsort

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

```
void qsort(void *base, size_t nel, size_t width,
int (*compar)(const void *, const void *));
```

enkelt, logisk

qsort

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

```
void qsort(void *base, size_t nel, size_t width,
int (*compar)(const void *, const void *));
```

enkelt, logisk og lett leselig.

qsort

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

```
void qsort(void *base, size_t nel, size_t width,
int (*compar)(const void *, const void *));
```

enkelt, logisk og lett leselig. tar inn startpunkt, antall, bredden på elementene

qsort

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

```
void qsort(void *base, size_t nel, size_t width,
int (*compar)(const void *, const void *));
```

enkelt, logisk og lett leselig. tar inn startpunkt, antall, bredden på elementene ... og påkaller ondskapen selv

Outline

1. Comparison between Languages
   - Python and C
   - Java and C
2. Why C
   - Is there any pros at all?
   - Funksjonskall
3. IO
   - IO functions
   - `printf()` example
   - `scanf()` example
4. Minnehåndtering
   - pekere
   - struct
5. Sortering
   - Funksjonspekere

Funksjonspekere

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

- Ingen lambdafunksjoner.
- Ingen sammenligningsoperatorer eller funksjoner innebygd i objektene.
- bare funksjoner

Funksjonspekere

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

- Ingen lambdafunksjoner.
- Ingen sammenligningsoperatorer eller funksjoner innebygd i objektene.
- bare funksjoner som vi kan ta adressen til

hva må compar() gjøre?

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

- parametre, to pekere
- returnerer et tall

strcmp

Introduction to C
Torje Hoås Digernes
Comparison between Languages
Why C
IO
Minnehåndtering
Sortering
Funksjonspekere

```
int (*compar)(const void *, const void *);
int   compar (const void *, const void *);

int strcmp(const char *s1, const char *s2);
```

Men vi kan bruke strcmp

Introduction to C

Torje Hoås Digernes

Comparison between Languages

Why C

IO

Minnehåndtering

Sortering

Funksjonspekere

```
int strcmp_indirect( void **a, void **b){
    return strcmp((char*)*a,(char*)*b);
}
```

endelig sortering

Introduction to C
Torje Hoås Digernes
Comparison between Languages
Why C
IO
Minnehåndtering
Sortering
Funksjonspekere

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int strcmp_indirect( char **a, char **b){
    return strcmp(*a,*b);
}

int main(){
    char *words[]={"Iä!","shub-niggurath!","the","Black",
        "Goat","in","the","Woods","with","a","Thousand",
        "Young",NULL};
    size_t length= 0;
    while(NULL!=words[length]){length++;}
    qsort(words,length,sizeof(char*),strcmp_indirect);
    for (size_t index = 0; index < length; index++) {
        printf("%s ",words[index]);
    }
}
```