

Formål:

Hvor er vi i IT-verdenen? Hvad er JAVA? – lidt om hardware og koder

At oprette en **Klasse** med **Attributter** og udskrive på Konsollen

Emner: Klasser, Attributter og Objekter

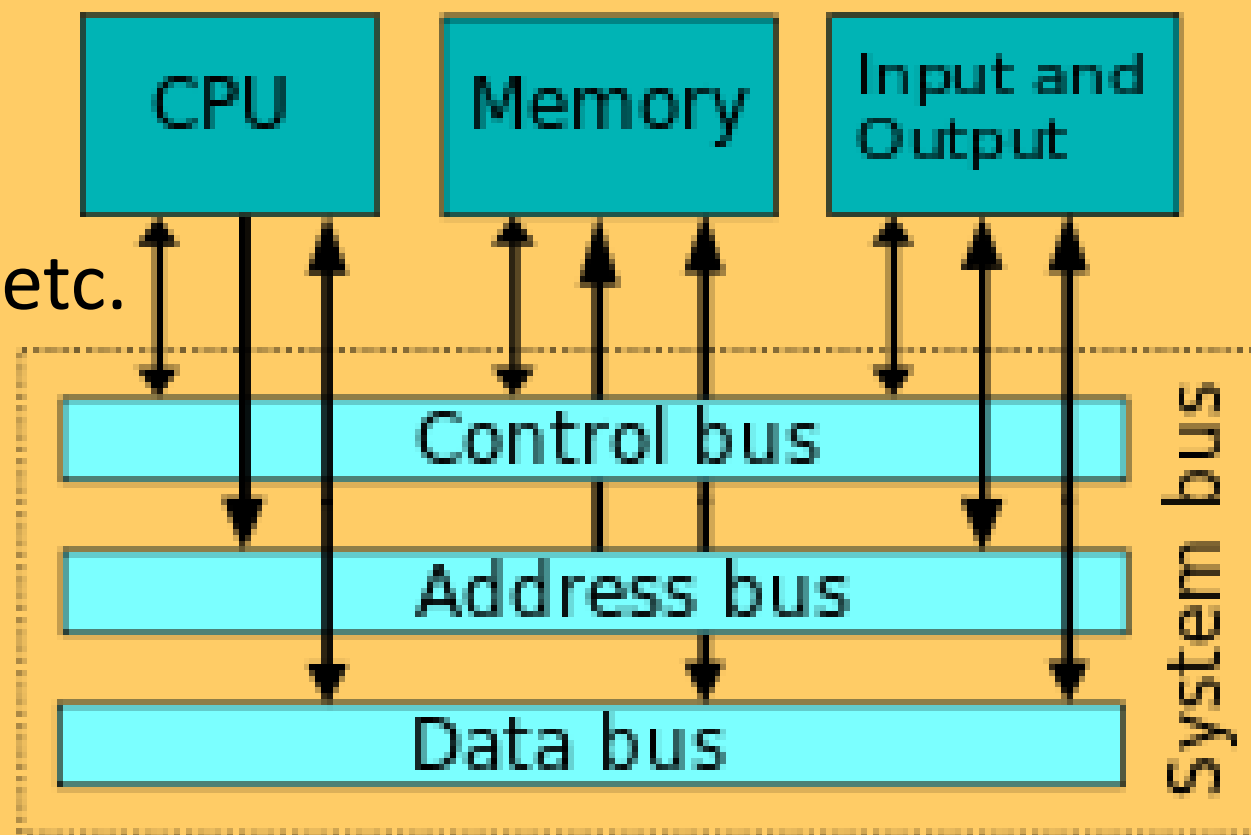
Litteratur: Reges: Kap.: 8.1 - 8.2

Reges: Kap.: 2.1 - 2.2

Hvad er en computer?



- CPU ? Clockfrekvens, Clocks per instruction, RISC?
- CPU-Typer, instruktionssæt, Rambussen, Databussen – 64bit
- Ram ?
- Hdd?
- Drev, grafikkort, USB-stik etc.

























Kode

- Hvad er kode?
- Hvad er et OS?
- Hvad er en fil? .txt .bat .com. Exe osv...
- Hvad er en compiler? Hvad er en interpreter (fortolker)
- Hvad er maskinkode? Højnivauprogrammering?
Fx: Algol, Turbo Pascal, Delphi , C++, PHP, C# (Af danske IT-udviklere 😊)

SWK: Kunsten at programmere

- JAVA
 - JAVA er et af verdens mest benyttede kodesprog
 - Platforms uafhængig pga pseudo maskinkode

HUSK:
I er studerende! Væn jer til kildekritik – vær opmærksomme på aktualitet og kvalitet når i arbejder med antagelser og vurderinger som i lægger til grund for jeres professionelle vurderinger og råd 😊

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

Kilde: IEEE Spectrum <https://Spectrum.IEE.ORG> Kilde og metoder kan ses fra deres site!

BEMÆRK: Selv om vi anvender JAVA er det ikke det der er fokus!!

Vi kigger på det at kode – sproget er irrelevant – JAVA er valgt både fordi det er meget udbredt og fordi det er alsidigt og rummer de egenskaber og principper som de øvrige sprog i vil støde på

Om undervisningen . . .

Læs op ad 2 omgange !!!

- **INDEN:**

Læs pensum fra **semesterplanen** fra gang til gang

Læs og forstå – er det svært så skim det – læs overskrifter forstå hvad det handler om så du kan stille gode spørgsmål under gennemgangen

- **UNDER:**

Følg med og grib dagens emne og husk du skal **lære det , forstå det og kunne arbejde med det!**

Følg med på de slides vi gennemgår – **Tag noter og spørg – Vær nysgerrig 😊**

- **EFTER:**

Gennemgå pensum igen – nu med læringen som ballast. **Renskriv noter** og gennemfør pkt.

INDEN undervisningen for næste gang 😊

BEMÆRK: Det er **VIGTIGT** at man skriver koden selv – det er OGSÅ et håndværk og ikke kun en mental udfordring 😊 MEN det er super givtigt at arbejde i grupper 2-4 er ideelt – find læsemakkere hurtigt – det giver rigtig meget at diskutere og tale om de nye begreber i tilegner jer 😊

Studiestartsprøve og Eksamen

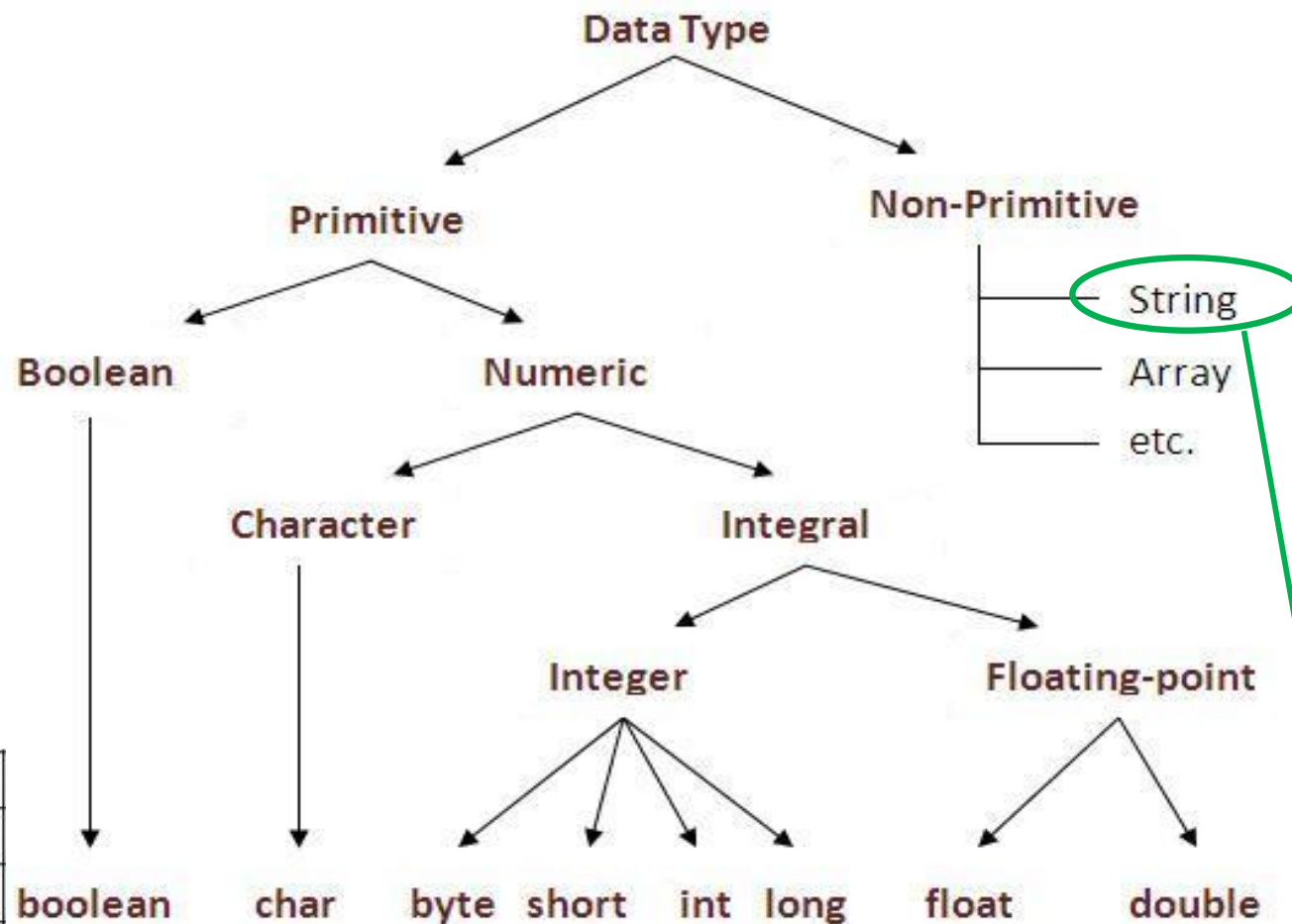
- Efter fem uger kommer studiestartsprøven...
- Efter semesteret skal i til eksamen i Konstruktion



Klasse ex.:

```
public class Car{  
  
    String brand = "Kia";  
    String model = "C'eed";  
    int year = 2016;  
  
    public void drive(){  
        System.out.println("Jeg er en bil der kører!");  
    }  
}
```

Data Typer



	Size	Default Value	Type of Value Stored
byte	1 byte	0	Integral
short	2 byte	0	Integral
int	4 byte	0	Integral
long	8 byte	0L	Integral
char	2 byte	'\u0000'	Character
float	4 byte	0.0f	Decimal
double	8 byte	0.0d	Decimal
boolean	1 bit (till JDK 1.3 it uses 1 byte)	false	True or False

Vi har snydt lidt og allerede taget en Non-Primitive type i brug – Og brugt indbyggede features som fx ToString-metoden. (LÆS I BOGEN !! 😊)

Klasse - Hovedprogrammet

```
public class Broker{  
  
    public static void main(String[] args){  
  
        Car minBil = new Car();  
  
        System.out.println("Mærke: " + minBil.brand);  
        System.out.println("Model: " + minBil.model);  
        System.out.println("Årgang: " + minBil.year);  
  
        //minBil.drive();  
  
    }  
}
```

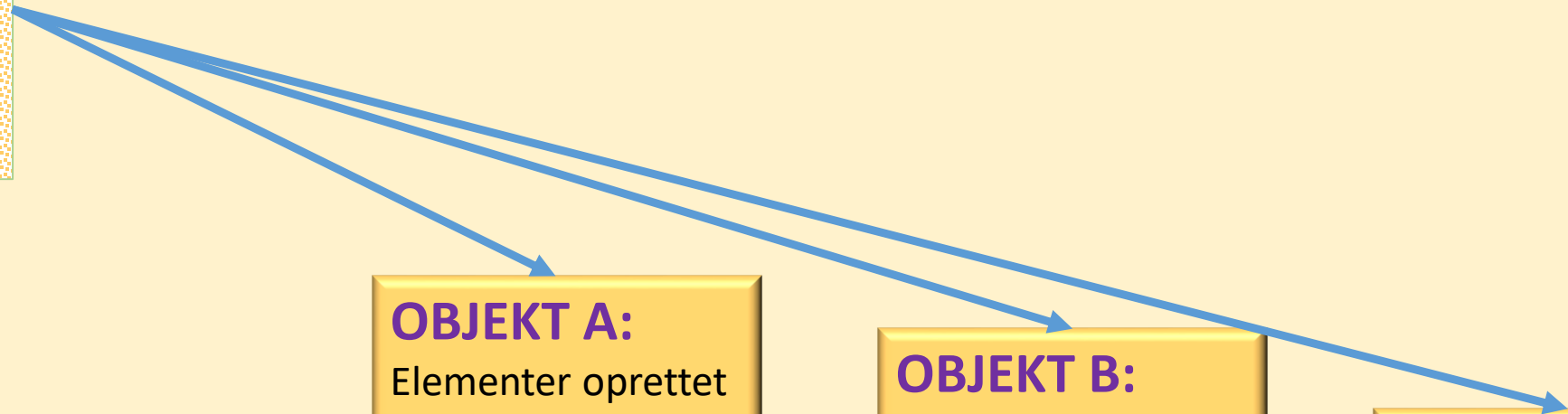
Klasse metoder!

Metoder som eksisterer UDEN for objektet – og kan kaldes UDEN et objekt er oprettet

Klasser & Objekter

KLASSEN:

Skabelon
og
instrukser



OBJEKT A:

Elementer oprettet
efter skabelon!

A kontrolleres og
bearbejdes med
instrukserne som
fulgte med klassen

OBJEKT B:

Elementer oprettet
efter skabelon!

A kontrolleres og
bearbejdes med
instrukserne som
fulgte med klassen

OBJEKT C:

Elementer oprettet
efter skabelon!

A kontrolleres og
bearbejdes med
instrukserne som
fulgte med klassen

God programmeringsstil:

Vi ønsker KUN at tilgå vores objekter
gennem metodekald!

Hvorfor egentlig det???

Øvelse: "Motorbike"

- Kod et Java projekt med en klasse for broker og en for motorcykel
 1. Hvilke fields skal motorcyklen have?
 2. Udskriv fields i konsollen (gerne via en metode ☺)
 3. Gennemgå og forklar din kode for sidemanden! (hvad gør koden, linie for linie)
 4. Forklar i klassen hvad der foregår i koden – redegør for (noget) af nedenstående
- Hvad er en klasse?
- Hvad er et objekt?
- Hvad er primitive typer, og hvordan og hvornår bruger vi dem?
- Hvad er JAVA? Hvad vil det sige at 'compile'?
- Kan du forklare forskellen på fortolkere og Compilere?
Og hvad er det nu lige JAVA gør? Er det en 'rigtig' compiler?

Formål:

At oprette en **Metode** og forstå konceptet og kunne forklare det.

Hvordan Objekter tilgås på den rigtige måde. Vi undersøger begreberne og sikrer at vi forstår dem og kan forklare dem for hinanden!

Emner:

Metoder

Litteratur:

Reges: Kap.: 8.1 - 8.2

NÆSTE GANG
Modul 2