# Objekt-Orienteret Programmering

## Modul 0

Introduktion osv.

## Modul 1.1 – Primitive Typer

### Part 0: Hello, World

#### Klasser

Java programmer består af én eller flere "klasser". Hver klasse repræsenterer nogetfunktionalitet.

Hver klasse er i udgangspunktet gemt i en fil (der er navngivet efter klassen):

Klassen Person er gemt i filenPerson.java

Funktionalitet i en klasse gøres tilgængeligt igennem metoder.

Hvis en klasse har en main metode kan den afvikles.

Lad os tage et eksempel . . .

#### Hello World

Public class HelloWorld

{

Public static void main (String[] args) {

System.out.print(“Hello,World”);

}

}

#### Fra Kildekode til Afvikling

Hvad sker der egentlig, når I afvikler jeres javaprogram?

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, diagram

Automatisk genereret beskrivelse

### Part 1: Datatyper

#### Primitive Datatyper

Datatyper bruges til at beskrive ”typer af data”.

Data er ofte noget der bliver produceret (eller ændret) når vi kører et program. Vi bruger typen til at beskrive rummet af gyldige værdier for data.

Java har 8 primitive datatyper. Det er de første typer vi skal beskæftige os med.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse

Det skal lige siges at en fuld unicode tegn fylder 32 bit men Java har en work around

#### Datatype Øvelser

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, linje/række

Automatisk genereret beskrivelse

### Part 2: Variable

#### Variable 🡪 Definition

## Modul 1.2

## Modul 1.3

## Modul 1.4 – Metoder

### Part 0: Funktioner i Matematik