# Begrebsøvelse programmering C – Karla

1. **Compile (compilering)**

Når man compiler et program gør man det læseligt for en almindelig computer. I processing vil det sige at man laver det om til Java, så en almindelig computer kan læse og køre det program man har skrevet.

1. **Syntaks**

Den rækkefølge tingene sker i et program. Først sker dette derefter sker dette osv.

1. **Variable**

En variable er et sted hvor der ligger noget data man kan hente i sit program. Der findes globale og lokale variabler og man kan lave sine egne.

1. **Datatype**

Datatype handler om hvilken slags data noget pågældende data er. Er det tal, pixels eller noget tredje.

1. **Float**

Float er en datatype der er decimaltal

1. **Int**

Int er et stort helt tal

1. **Char**

En datatype der indeholder characters altså enkelte bogstaver såsom ’a’

1. **Boolean**

En datatype der enten er true eller false. Kan bruges når der skal foregå beslutninger i et program

1. **Debugging**

Det arbejde man laver efter man kører et program og det ikke helt virker som det skal. Her går man sin kode igennem og finder ud af hvad der er galt med det.

1. **If sætning**

Denne sætning bruges hvis man gerne vil have programmet til at gøre noget bestemt, hvis noget andet bestemt sker. Det kunne f.eks. være: hvis knappen trykkes på skal den blive grøn

1. **For loop**

Et for loop bruges til at sige at noget bestemt skal ske og blive ved med at ske indtil noget andet bestemt holder op med at ske. Det kunne f.eks. være at knappen skal være grøn når der trykkes på knappen og blive ved med at være grøn indtil der trykkes på en anden knap.

1. **Increase/Decrease indent**

Indent betyder undrykning. I processing betyder de forskellige indrykningsgrader (hvor langt teksten er rykket ind) hvor i scope det hører til. Processing gør det af sig selv ved tryk på ctrl+t. Disse indryk giver et overblik over hvad for noget kode der hører sammen.

1. **Placering af tuborg-parenteser**

Tuborgklammer viser hvor en blok kode starter og hvor den slutter igen. Der kan godt være flere blokke af kode inden i hinanden. F.eks. er void draw() en blok af kode, men der er mange andre blokke inden i det.

1. **Kode kommentar**

Man bruger kodekommentarer til at skrive hvad en linje kode gør i hverdagssprog sådan at man kan huske det næste gang man kigger på sin kode. Dte hjælper også andre med at forstå ens kode. Man laver kommentarer ved at lave // og skrive derefter.

1. **Pseudo kode**

Pseudo kode er kode der ikke ville fungere i programmet. Det er ikke skrevet i programmeringssproget, men det viser hvad programmet skal kunne og hvordan det fungerer uden at man reelt har lavet programmet. Det kan bruges til at få overblik over og organisere ens program.

1. **Flow chart**

Et flowchart viser programmets syntaks på en overskuelig måde. Man bruger det til at skabe overblik over de ting der sker i ens program

1. **Function**

En funktion er i essens et lille program inden i et program. Man kalder en funktion og den gør noget bestemt med det data man giver den. Funktionen kunne f.eks. hedde 2. gradsligning og kunne dermed løse en andengradsligning man gav den og dette ville kunne bruges i ens program.

1. **Input parameter**

Det er den data man putter ind i en funktion for at få den til at gøre noget bestemt. Det kunne f.eks. være variabler.

1. **Output(return) parameter**

Det er den data man får ud af en funktion efter at den har udfært sit arbejde.

1. **Global variable**

En global variable ligger i det yderste scope og kan derfor ses og bruges af hele programmet.

1. **Lokal variable**

En lokal variable hører til en bestemt blok af kode og kan ikke tilgås uden for denne.

1. **OOP(Objekt Orienterede Programmering)**

OOP går ud på at man programmerer efter at lave objekter og klasser man derefter kan bruge i sit program. Det kunne f.eks. være at man ville lave et program hvor biler kørte. Så ville man lave en klasse der hed bil og deri lave alle de forskellige biler der skulle køre.

1. **Class(Klasse)**

En klasse er det overordnede ”navn” på et objekt. Det kunne f.eks. være klassen bil hvor der under den var objekterne blå bil og rød bil osv.

1. **Object**

Et objekt er så en del af en klasse. Klassen giver de overordnede parametre og derefter kan objekterne designes ud fra det.

1. **Constructor**

I constructoren ”Bygger” man objekter udfra klassen. Klassen fungerer som en opskrift der tager imod ingredienser (data) og consstructoren er ovnen hvor disse ingredienser bages sammen til et objekt.