

# Datastruktur och Algoritmer - Föreläsning 1

## What algorithm?

→ An algorithm in Mathematics and Computer Science is a limited set of well-defined instructions to solve a problem which from given starting states with certainty leads to specific end states.

A problem with a specific input is called an **instance of a problem**. An algorithm is correct if the algorithm produces steps for each problem instance and returns the required output. **The algorithm solves the problem**

→ Important for Internet, Computers, Security, Biology, Social networks and Physics

=> **Focus on Data-Structures and not if the code works**

## Terms in Basic Java

Primitive data types - **int double boolean char** - A set of values and a set of operations on those values

Identifiers - **a, abc, Ab, a-b, ab123** - A sequence of letters, digits and **\_**, the first of which is not a digit

Variables - [any identifier] - Names a data-type value

Operator - **+, -, \*, /** - Plus, minus, multiply and division on Java

Literal - **int, double, boolean, char** - Source-code representation of a value

Expression - **int:  $10 + (11 - 10) / 2$**  - A literal, a variable or a sequence of operations on literals and/or variables that produces a value  
**double:  $1.0E-15 * e$**

Declaration - **int i;**  
**double c;** - Create a variable of a specific type named with a given identifier

Assignment - **a = b + 3** - Assign a data-type value to a variable

Initializing declaration - **int i;**  
**i = 1;** - Increases the value after iteration

conditional (if) - **if(x < 0) x = -x;** - Execute a statement depending on the boolean expression

## Static methods and Methods

Public      Static      double      sqrt      (double c)      argument variable  
Signature      Return type      method name      Argument type

**E** **if(c < 0) return Double.NaN;**

**double err = 1E-15;**

**double E = c;**

→ **err and E = local variables**

**while (Math.abs(E - (c/E)) > err \* E)**

→ **Method body**

**E = (c/E + E) / 2.0;**

→ **another method**

**return E;** - Return statement

**Arguments:** ~~Pass-by-Value~~ - Primitive type  
Pass-by-Reference - (Object, array)

→ Methods can only return one thing, but they may have many return statements

→ Method names may be overloaded

↳ `Math.min(int x, int y)`, `Math.min(double x, double y)` of an object, order elements in an array.

→ Without extra code, a Java program may access input via:

1. Command Line arguments

↳ `public void static main(String[] args)`

2. Environment Variables

↳ `Java-Djava.library.path=/home/jim/libs-Jar MyProgram.jar`

3. Standard input stream (stdin)

↳ An abstract stream of characters

→ A Java program can write to

1. → Standard output stream (stdout)

2. (stderr)

## APIs and Object oriented Programming

→ Object oriented design

↳ Abstract data types

(Contract)

→ An **Application Programming Interface (API)** is an interface defining the behaviour of an abstract data-type (ADT)

An API encapsulates the behaviour/implementation of an abstract data-type (ADT)

→ The client doesn't need to know anything of the internal implementation of the ADT

→ Class

↳ Templates for objects

↳ Class Methods (only one instance in the class)

↳ Class Variables (only one instance in the class)

Instance/object

→ Objects are instantiated from the class

→ Can access methods and variables

→ Each instance has its own set of instance methods and variables

Public class Counter

Counter(String id) - Create a counter named Id  
Void increment - Increment the counter by one  
int fully() - Number of increments since creation  
String toString() - String representation

→ Create/instantiate an object  
Counter heads = new Counter("heads")  
Declaration to associated with | Call on constructor to create an object  
Invoke Method  
heads.fully() - tails.fully()

## API Design - String Class

String() - Create an empty string  
int length() - Length of the string  
int charAt(int i) - I-th character  
int indexOf(String P) - First Occurrence of P  
int indexOf(String P, int i) - First Occurrence of P after i  
String concat(String E) - This string with E - Appended  
String substring(int i, int j) - Substring of this string (ith to j-th - 1st char)  
String[] split(String delim) - Strings between occurrences of delim  
int compareTo(String E) - String Comparison  
Boolean equals(String E) - is this string's value the same as E?  
int hashCode() - Hash Code