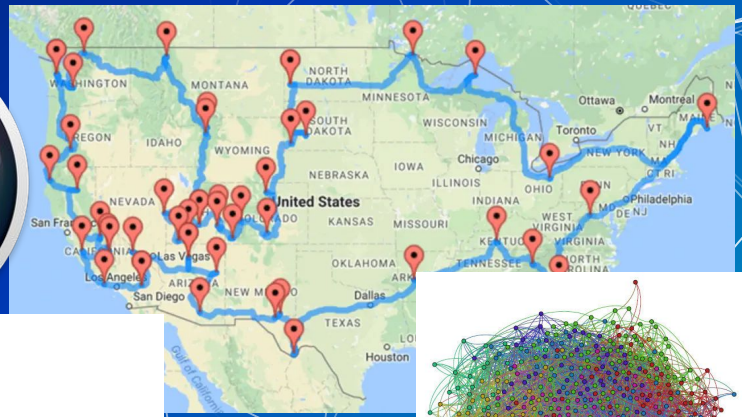
Introduction to Data Structures and Algorithms

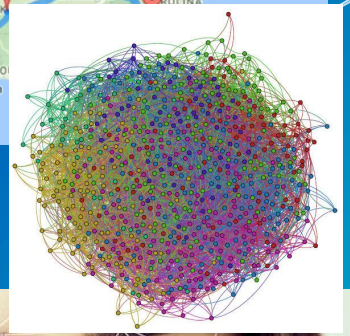
> Bad programmers worry about the code. Good programmers worry about the data structures and their relationships
>
> -Linus Torvalds

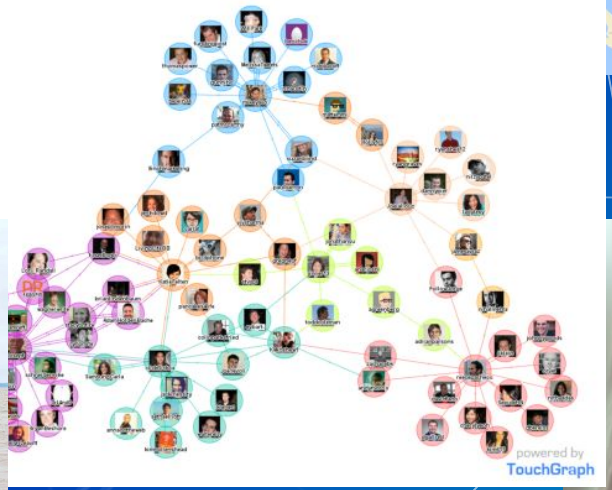
1. What are DS&A

Data Structures and Algos Intro

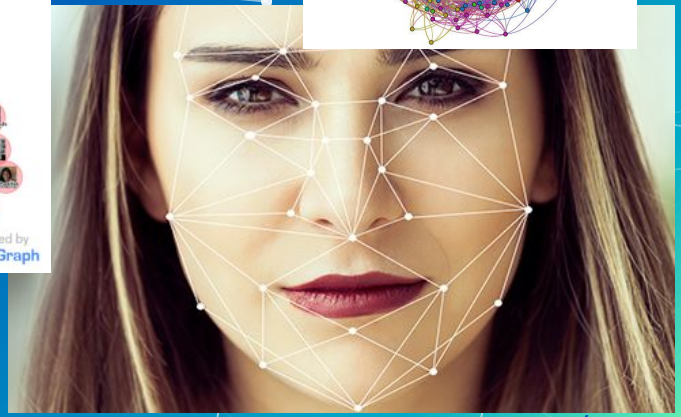

Autonomous Mode


United States
Ottawa
Montreal
Toronto
Chicago
Philadelphia
San Diego
Dallas
Houston
Las Vegas




powered by
TouchGraph

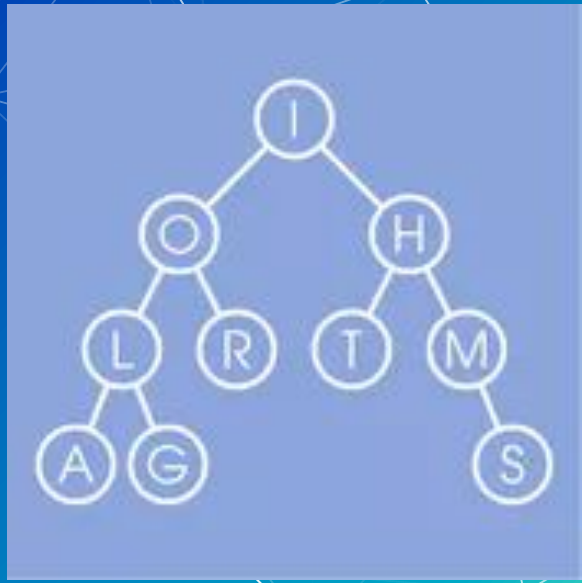

kite: 97%
kite: 85%
kite: 82%
kite: 90%
kite: 54%
kite: 92%
person: 97%
person: 86%
person: 76%
person: 54%
person: 99%
person: 99%



IBM

Impact of DS&A

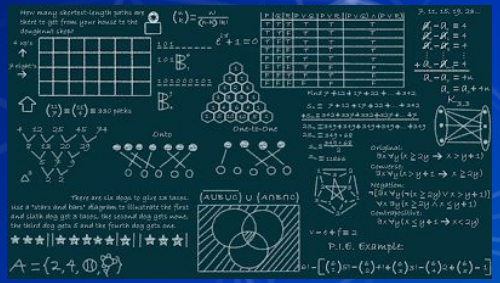
- Heart of computer science
  - CS revolves around data structure and algorithms
- Huge scope for solutions in real world applications/probs
- Theoretical cs ⟶ serves as basis for DS&A applications



DS&A - What is the difference?

Data Structures -

- Organize and collect data through efficient means from which we can perform various algorithms/operations



Algorithms -

- Operations or set of instructions applied on a data structure to solve a particular problem → real world solution to application

DS&A Examples

Data Structures -

- Arrays & Strings
- Stack
- Queue
- List
- Hashing + Hash Tables, etc.
- Heaps
- Graph
- Trees

Algorithms -

- Sorting & Searching algos
- Stacks & Queues Algos
- Hashing Algos
- Greedy Algorithms
- Recursion
- Dynamic Programming
- Graph Algos
  - Basic (DFS, BFS, etc.)
  - Advanced

2. Java + IDE

Data Structures and Algos Intro

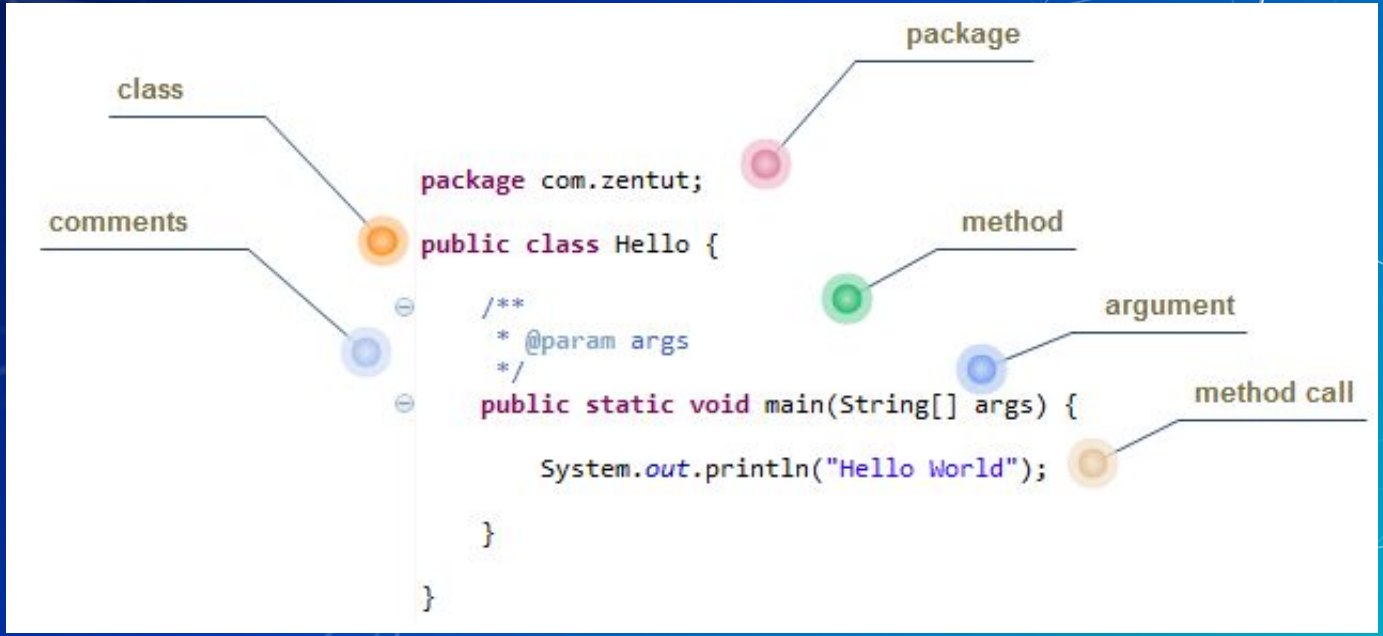
What is Java?

- Very popular and widely used
- Object oriented based
- Class based
- Main language we will be using
- Presentations → theoretical based, pseudocode
  - Github → contain code

What is IDE?

- Integrated Development Environment
  - Eclipse
  - IntelliJ
  - NetBeans
  - BlueJ
- Write code, debug, iterative process

Components of Java Program



Primitive Data Types

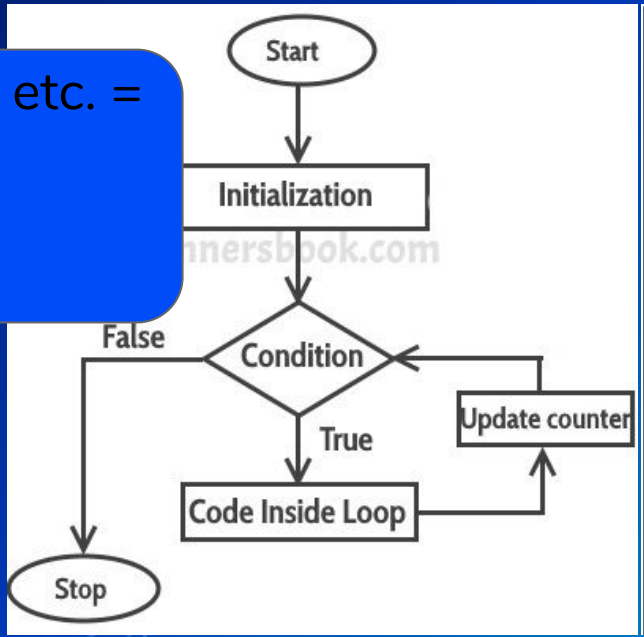
Type	Description	Default	Size	Example Literals
boolean	true or false	false	1 bit	`true, false`
byte	twos complement integer	0	8 bits	(none)
char	Unicode character	`\u0000`	16 bits	`'a', '\u0041', '\101', '\\', '\'', '\n', 'ß'`
short	twos complement integer	0	16 bits	(none)
int	twos complement integer	0	32 bits	`-2, -1, 0, 1, 2`
long	twos complement integer	0	64 bits	`-2L, -1L, 0L, 1L, 2L`
float	IEEE 754 floating point	0.0	32 bits	`1.23e100f, -1.23e-100f, .3f, 3.14F`
double	IEEE 754 floating point	0.0	64 bits	`1.23456e300d, -1.23456e-300d, 1e1d`

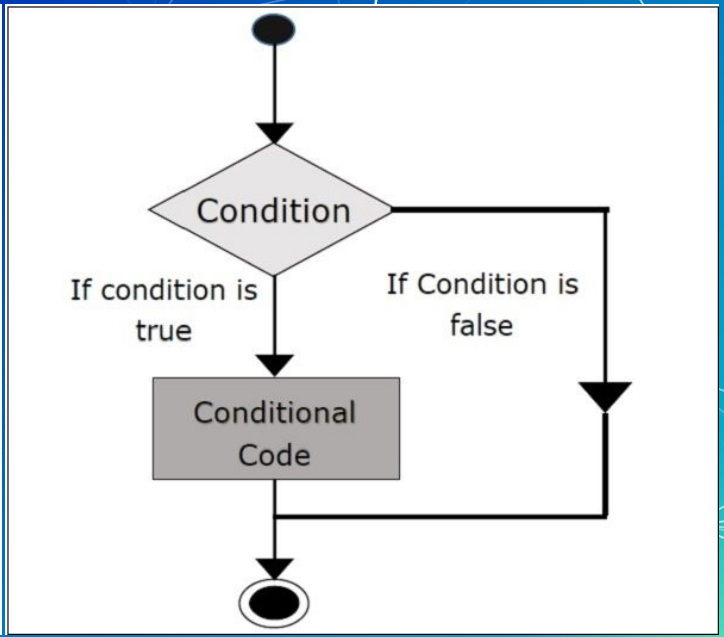
Data Types used as building blocks

If Statements and For loops

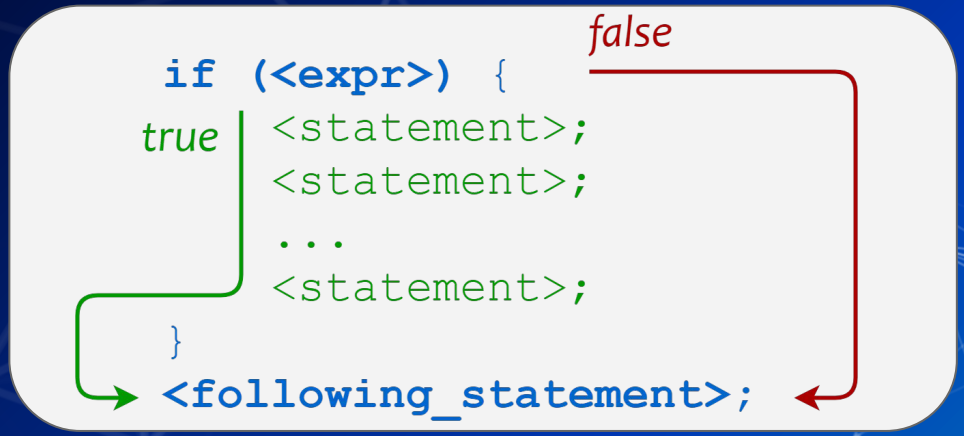
If statement, if-else, etc. = conditional

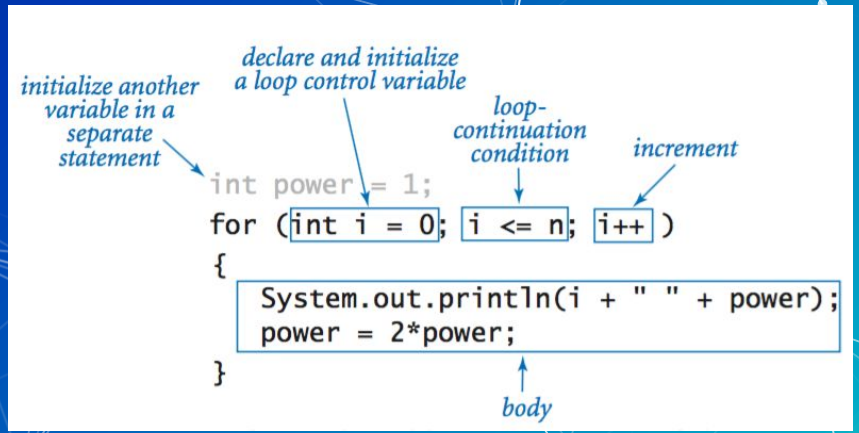
For loop = iterative





If Statements and For loops





3. Basic Data Structures

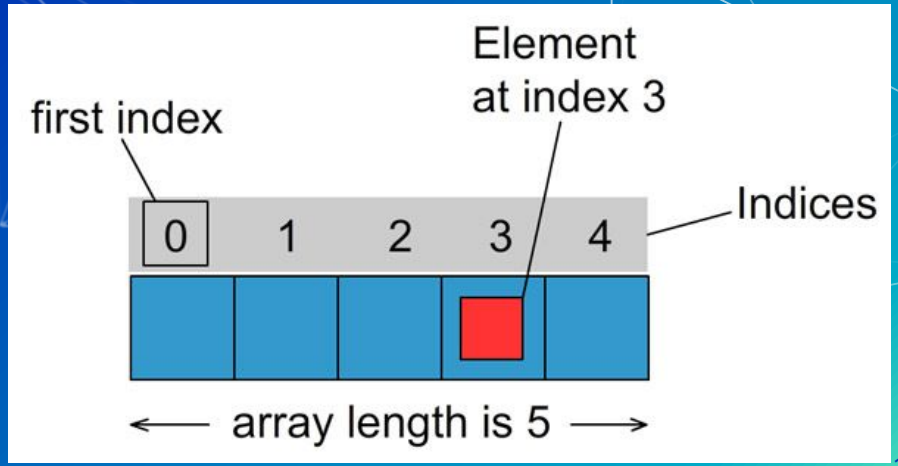
Data Structures - Arrays, Strings

Array

Ordered arrangement

1. Starts with index 0
2. Array length = n
3. Fixed size
4. n-1 indices

Ex - List of students, etc.



Initializing an Array

JavaArrayInitialization.java

```
1  public class JavaArrayInitialization {
2
3      // examples for initializing an array in java
4      public static void main(String[] args) {
5          // initialize primitive one dimensional array
6          int[] arrInt = new int[5];
7
8          // initialize Object one dimensional array
9          String[] strArr; // declaration
10
11         strArr = new String[4]; // initialization
12
13         // initialize multidimensional array
14         int[][] twoArrInt = new int[4][5];
15
16         // multidimensional array initialization with only leftmost dimension
17         int[][] twoIntArr = new int[2][];
18         // complete initialization is required before we use the array
19         twoIntArr[0] = new int[2];
20         twoIntArr[1] = new int[3];
21
22         // array initialization using shortcut syntax
23         int[] arrI = { 1, 2, 3 };
24         int[][] arrI2 = { { 1, 2 }, { 1, 2, 3 } };
25
26         int[] twoArrInt1[] = new int[4][5];
27
28         int twoIntArr2[][] = new int[5][];
29     }
30
31 }
```

Declaring vs. Initializing

- Declaring an array = declaring it exists, allocates some storage for the array

- Initializing = will contain “new” keyword, specifies value object will store (assigning)

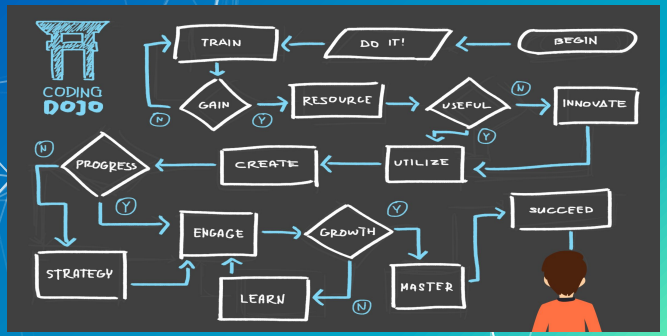
String

Sequence of characters

1. Immutable
   a. Once created ⇒ Can't be changed
2. String length = n
   a. Not fixed size
      i. Don't declare before deciding what sequence itself will be
3. Index starts at 0
4. Cannot apply any mathematical operations that can do on int, double,etc.l

4. Basic Algos

Algos - Arrays & Strings algos + if-statements & Algorithmic Complexity Overview



https://www.youtube.com/watch?v=H_aLU-NOdHM

Array Algo 1 - Print elements

Array Algo 2 - Linear Search for Element

Array Algo 3 - Compute Average of Elements

Array Algo 4 - Find the max and min elements

Array Algo 5 - Swapping elements between 2 arrays

Strings Algo 1 - Implement length() method

Strings Algo 2 - Implement indexOf() method

Strings Algo 3 - Implement matches()

Strings Algo 4 - Substring search (Ex - search number of times “cat” appears in “clsatcat”

Strings Algo 5 - Replace all substrings with newchar (implement replaceAll())

Readings Online - Github

- Will post content on this lecture
  - Objects
  - String and Array algorithms discussed in class
    - Pseudocode or Java
  - Additional introductory-level java concepts needed
- Will post content for next lecture
  - Searching and Sorting algorithms

See You Next Session!