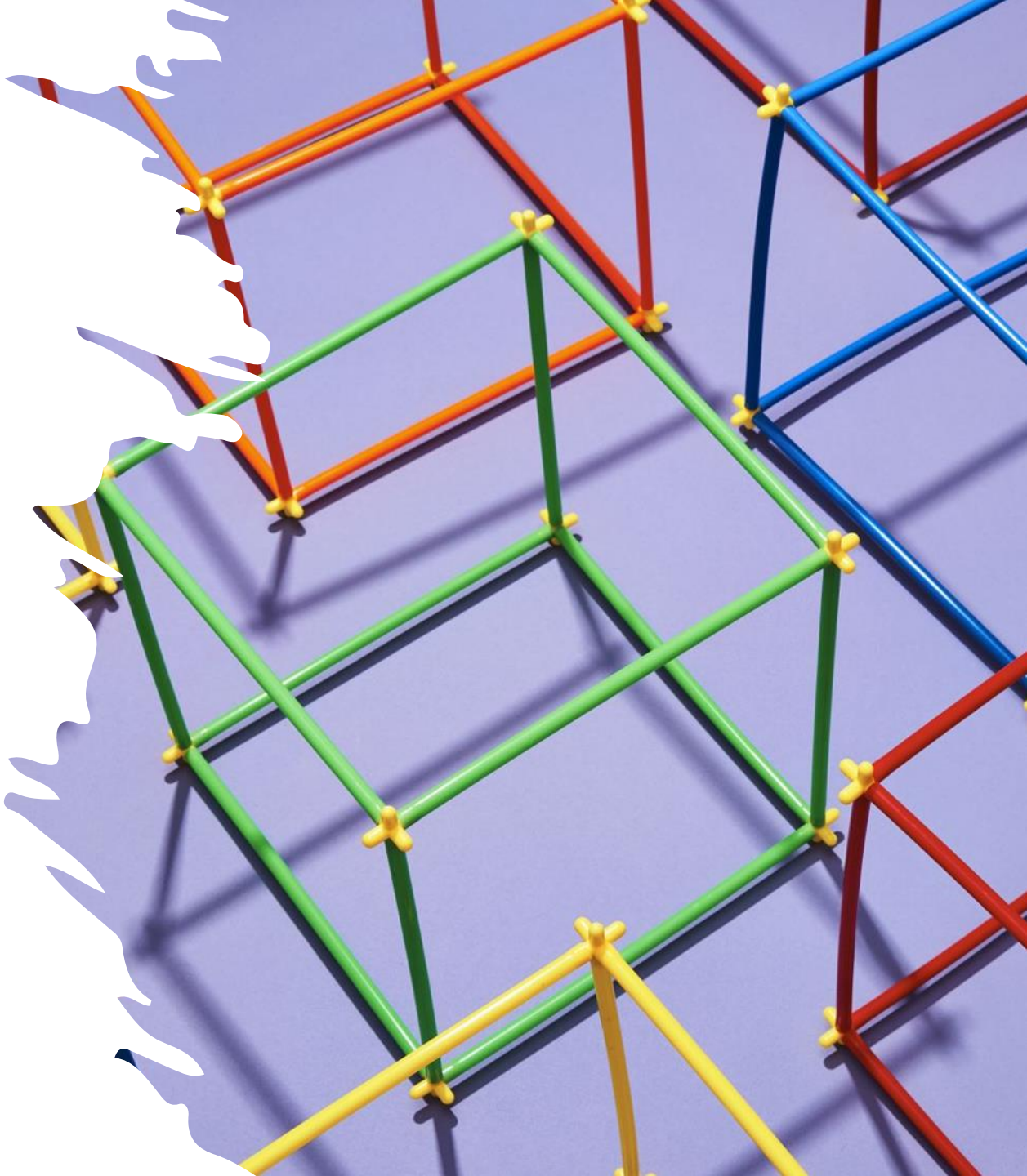
Data Structure
Lecture 1: Introduction

Prepared by

Dr. Mohammed Salah Al-Obiadi



Variables and Data Types

- **Variable** is any entity that can take on different values.
- Consider the below equation:

$$x^2 + 2y - 2 = 1$$

- This equation has variables *x* and *y*, which hold values (data).
- **Data Type** is a set of data with predefined values.
- The variables *x* and y in the above equation can take any values such as
  - Integer numbers (10, 20), Real numbers (0.23, 5.5), or Boolean (0 or 1).
- There are two types of data types:
  - System-defined data types.
  - User-defined data types.

System-defined data types:

- These are the data types that are defined by system are called primitive data types.
  - Examples of such data types are int, float, char, double, bool, etc.
- Each data type has some bytes to store data.
  - For example:
    - int may take 2 bytes or 4 bytes of memory.
    - float may take 3 bytes or 4 bytes of memory.
    - char may take 1 byte of memory.
    - Symbols in ASCII codes like +, -, *, /, @,#, etc… take 1 byte of memory. (See Appendix A for the list of ASCII codes)
  - If we have x+y, and x is int and y is float. Assume int takes 2 bytes, and float takes 3 bytes, then the equation x+y take 2+1+3=6 bytes of memory.
    - Note that the symbol ‘+’ has 1 byte in the memory.

User defined data types

- If the system-defined data types are not enough, then most programming languages allow the users to define their own data types.
- Good examples of user defined data types are: structures in C/C + + and classes in Java.
- Here is an example of new defined data type called “newType”:

```
struct newType {
    int data1;
    float data 2;
    ...
    char data;
};
```

Data Structures

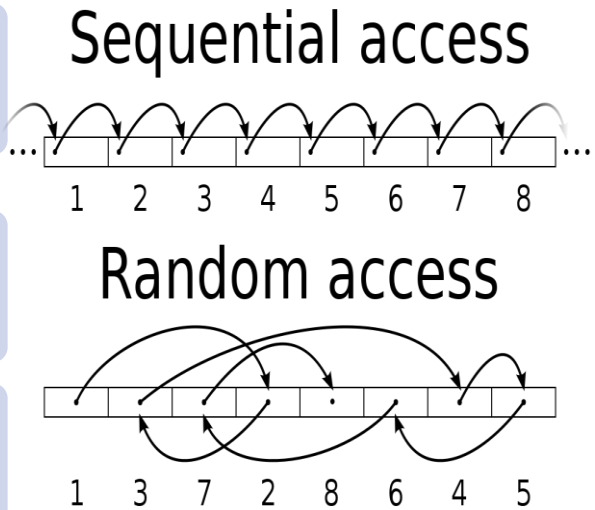
Data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently.

General data structure types include arrays, files, linked lists, stacks, queues, trees, graphs and so on.

Data structures are classified into two types:

1- Linear data structures: elements can be accessed in a sequential order (e.g. Arrays, Linked Lists, Stacks and Queues).

2- Non – linear data structures: elements can be accessed in a random order (e.g. Trees, tables, sets, graphs).



Operations Performed in Data Structure

1- Traversing

2- Insertion

3- Deletion

4- Merging

5- Sorting

6- Searching

Appendix A:

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]