

فوائد من أ.محمد أبو هدهود (سلسلة أساسيات مهمة لكل مبرمج)

خطوات تعلم البرمجة

(١) اطرء الناس السلبيين من حياتك

(٢) تعلم البرمجة وليس لغة برمجة

• الفرق بين البرمجة ولغة البرمجة

❖ **البرمجة** هي : أسلوب تفكير وتحليل منطقي للمشكلة لإيجاد أفضل الحلول

❖ **لغة البرمجة** هي : أداة لتطبيق حل المشكلة

- المبرمج هو : الذي يستطيع حل المشاكل
- لابد من تطوير المنطق أو التفكير لحل المشاكل ويحتاج لفترة طويلة
- إتقانك للغة البرمجة لا يعني أنك مبرمج

(٣) ابتعد عن لغات البرمجة الحديثة في البداية : لأنها لا تعلمك التفكير المنطقي

(٤) طور المنطق من خلال حل ٢٠٠٠ سؤال على لغة واحدة

❖ مثل C++ Labs , java Labs , python Labs تقريبا سنة أقل شيء

(٥) تعلم لغة من اللغات الحديثة مع Database وطبق مشروع عليه

❖ تعلم على قواعد البيانات Database شهرين

❖ مثال برنامج تسجيل في الجامعة مع تطويره

الدرس الأول: الفرق بين البيانات والمعلومات، والمعرفة والخبرة

الفرق بين البيانات (Data) والمعلومات (information)

البيانات هي: حقائق

١. حقائق
٢. غير مرتبة
٣. غير محللة
٤. غير معالجة
٥. ذات معنى قليل

المعلومات هي : البيانات + استنباط من البيانات

١. مشتقة من البيانات

٢. مرتبة

٣. محللة

٤. معالجة

٥. لها معنى

■ بيانات (Data) + تحليل (Analyses) = معلومة (Information)

■ معلومة + معلومات = معرفة (Knowledge)

■ معرفة + تطبيق (Applied) = خبرة (Wisdom)

الدرس الثاني: ما هو الكمبيوتر ، وما هي وظائفه ومكوناته ؟

الكمبيوتر هو: جهاز إلكتروني يشتغل ضمن أوامر يستقبل بيانات ويحلل البيانات بحسب قوانينك ويعطيك المعلومة ، يفهم فقط 0,1 وهو النظام الثنائي

وظائف الكمبيوتر (Functionalities of computer)

١. إدخال البيانات (Input Data)

٢. تخزين البيانات (Stores Data)

٣. معالجة البيانات (Process Data)

٤. إخراج البيانات (Output Info)

٥. التحكم في الوظائف (Control)

مكونات الكمبيوتر (Computer Components) اثنان:

❖ سافتيوير (Software)

❖ هاردوير (Hardware)

السوفتيوير هو: أي شيء غير ملموس في الكمبيوتر مثل البرامج، وينقسم الى قسمين

١. أنظمة التشغيل (System software) مثل ويندوز ماك windows

Unix , DOS , Linux , Android mac SO ,

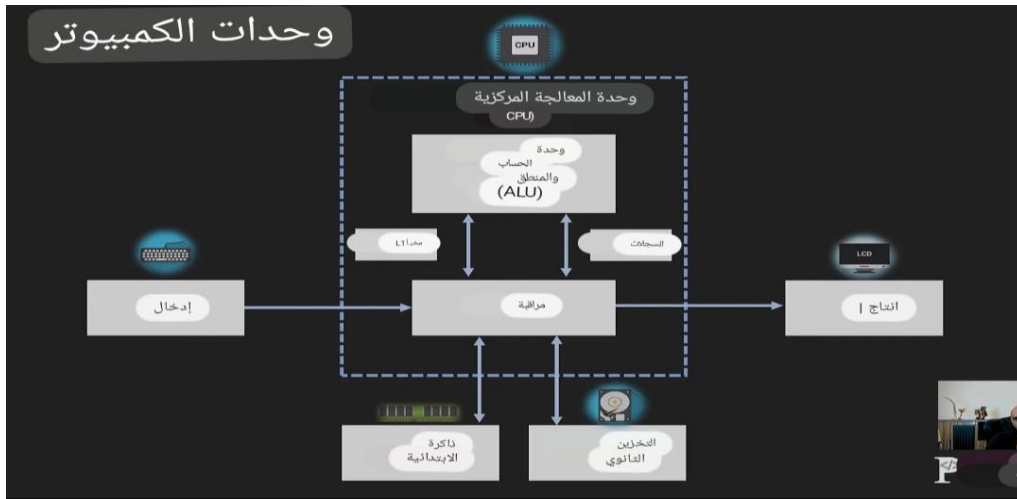
٢. التطبيقات (Application software) مثل وورد، كروم

[والمبرمج إما يبرمج أنظمة التشغيل أو التطبيقات]

هاردوير: أي شيء تستطيع لمسها مثل الهارد درايف، الشاشة، لوحة المفاتيح، RAM,

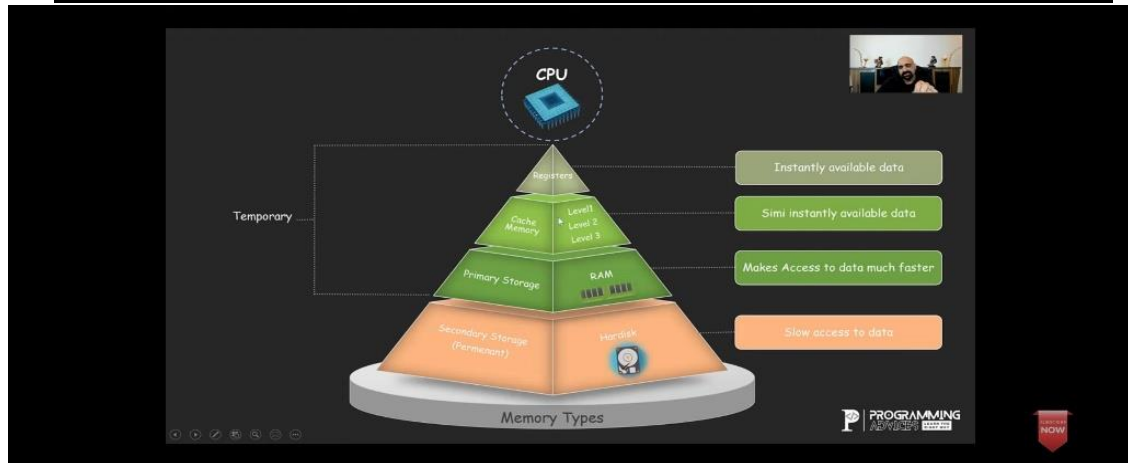
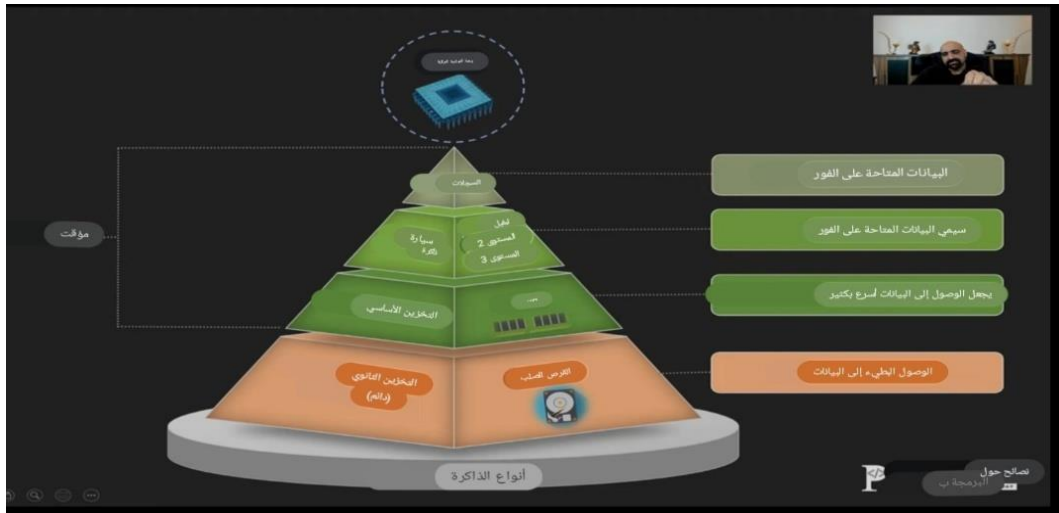
وحدات الكمبيوتر (Computer Units)

١. وحدات إدخال (Input) مثل الماوس، شاشة اللمس ، كيبورد
٢. وحدات إخراج (Output) طابعة ، شاشة الإخراج ، سماعات
٣. وحدة الذاكرة الرئيسية (Primary Memory) RAM & ROM
 - ❖ RAM الوصول الى المعلومة بشكل عشوائي ، وهو الاحتفاظ بالبيانات بشكل مؤقت للوصول السريع إليها كلما كانت مساحة (RAM) أكبر كلما كان أسرع
 - ❖ ROM وهي ذاكرة دائمة ، يحفظ فيها عادة الأوامر الرئيسية مثل التشغيل
٤. وحدة الذاكرة الثانوية (Secondary Storage) مثل الفلاش، هار درايف
٥. وحدة المعالجة المركزية (CPU) (Central Processing) وهي المتحكمة بكل الوظائف والعمليات الحسابية ، وهي العقل للكمبيوتر وهو مكون من (Transistor) وهو مفتاح كهربائي يسمح بمرور الكهرباء أو يمنع مرور الكهرباء أي ((0)(ON) أو ((1)(Off)) وهي اللغة التي يفهمها الكمبيوتر ، وهو مكون من عدة ملايين



أسرع وحدات التخزين من حيث تلقي الأوامر من CPU بالترتيب

١. Registers البيانات المتاحة على الفور
٢. Cache Memory الوصول الى البيانات بشكل مماثل أبطأ من الأول
٣. RAM Primary Memory الوصول الى البيانات بشكل أسرع أبطأ من الثاني
٤. Har disk Primary Memory (الذاكرة الدائمة) الوصول الى البيانات بشكل بطيء



وحدة معالجة الصور كرت الشاشة (Graphics Processing Unit (GPU))

وهذا يهم أصحاب الألعاب والمصممين

الدرس الثالث : النظام الثنائي، وتمثيل البيانات في الكمبيوتر

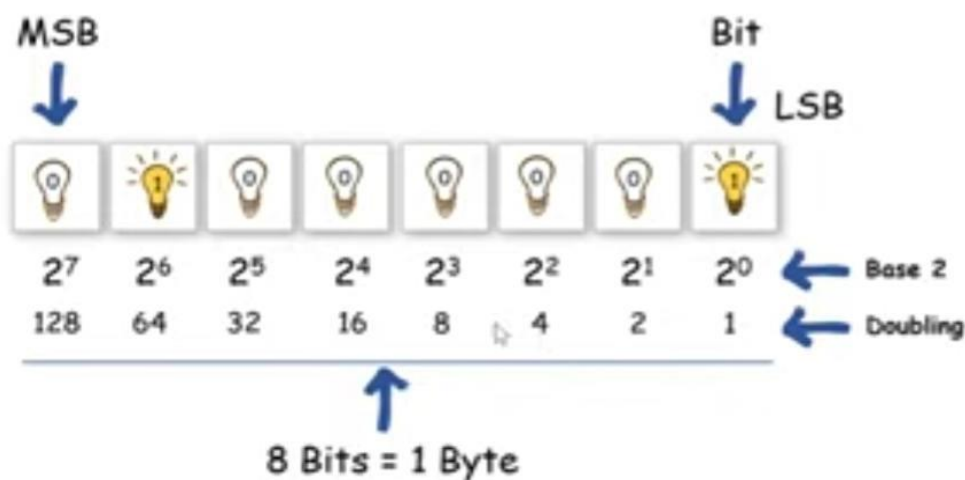
كيف تمثل البيانات في الكمبيوتر ؟

الفرق بين Decimal & Binary ؟

❖ النظام العشري Decimal System وهو مكون من عشر خانات من 0 الى 9

❖ النظام الثنائي Binary System يحتمل احتمالين إما 0 أو 1 فقط

- ما هو البت Bit ؟
- خانة صغيرة موجودة في الذاكرة يخزن فيها إما 0 أو 1 فقط
- ما هو Byte ؟
- هي أقل مساحة يمكن تخزين البيانات فيها ، وهو مكون من ثمان خانات من Bit
- أصغر خانة موجودة في Byte تسمى LSB ، وأكبر خانة تسمى MSB
- النظام الثنائي هو Base 2 لأن القاعدة تكون دائما 2 والقوة أي الأس هو الذي يزيد
- ميزة المضاعفة Doubling تخزين المعلومات كثيرة بأقل مساحة



أصغر وحدة في Byte وأكبر وحدة فيها أكبر								
	MSB			Byte = 65			LSB	
Binary	0	1	0	0	0	0	0	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal	128	64	32	16	8	4	2	1
	8 Bits = 1 Byte							

- Byte = 65 مجموع خانات $1 + 64 = 65$ (Decimal) النظام العشري
- يفهم الكمبيوتر $01000001 = 65$ (Binary) النظام الثنائي
- لغة الكمبيوتر هي Machine Language وهي (Binary)
- (Binary) يعطيك احتمال واحد لإيجاد الحل مثل 65
- Byte 1 = مجموع الأرقام (Decimal) $255 =$
- Byte 2 = مجموع الأرقام $65,535 =$ وهكذا

كيف يفهم الكمبيوتر الأحرف ؟

لكل حرف رقم خاص فيه (الحرف الصغير يختلف عن الحرف الكبير)

لا يفرق الكمبيوتر بين الحرف والرقم، بل يفهم بحسب السياق

- ❖ ASCII وهو لتمثيل الأحرف الإنجليزية فقط
- ❖ UNICODE وهو نظام عالمي للغات العالم ومنها اللغة العربية والرموز..

الفرق بين ASCII و UNICODE

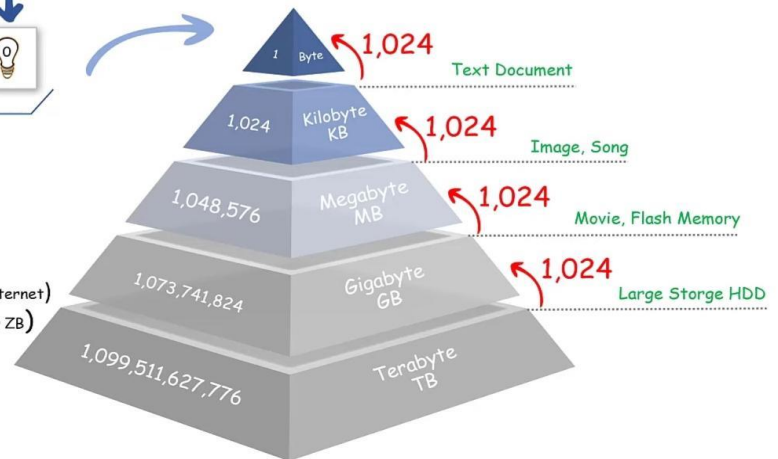
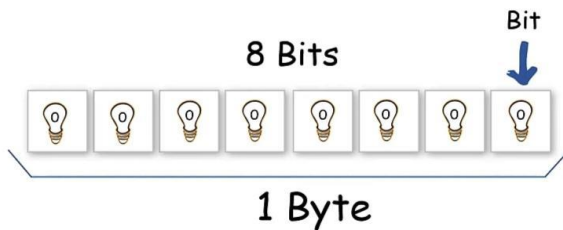
UNICODE	ASCII
عدد كبير من الأحرف مساحة أكبر	يدعم أحرف معينة مساحة أقل
له ٣ معايير	له معياران
٢٥٦	٧ بت = ١٢٨
٦٥,٥٣٦	٨ بت = ٢٥٦
أكثر من أربعة مليار	
يدعم أي لغة في العالم	لغة واحدة فقط الإنجليزية

Letter	Binary	Decimal	Letter	Binary	Decimal
A	01000001	65	a	01100001	97
B	01000010	66	b	01100010	98
C	01000011	67	c	01100011	99
D	01000100	68	d	01100100	100
E	01000101	69	e	01100101	101
F	01000110	70	f	01100110	102
G	01000111	71	g	01100111	103
H	01001000	72	h	01101000	104
I	01001001	73	i	01101001	105
J	01001010	74	j	01101010	106
K	01001011	75	k	01101011	107
L	01001100	76	l	01101100	108
M	01001101	77	m	01101101	109
N	01001110	78	n	01101110	110
O	01001111	79	o	01101111	111
P	01010000	80	p	01110000	112
Q	01010001	81	q	01110001	113
R	01010010	82	r	01110010	114
S	01010011	83	s	01110011	115
T	01010100	84	t	01110100	116
U	01010101	85	u	01110101	117
V	01010110	86	v	01110110	118
W	01010111	87	w	01110111	119
X	01011000	88	x	01111000	120
Y	01011001	89	y	01111001	121
Z	01011010	90	z	01111010	122

الدرس الرابع: السرعات ووحدات السرعة في الكمبيوتر
وحدات الذاكرة (Memory Units) من الأقل الى الأكبر

8 Bits = واحد بايت 1 Byte (١)
Byte 1024 = (KB) 1 Kilobyte (٢)
Kilobyte 1024 = (MB) 1 megabyte (٣)
Megabyte 1024 = (GB) 1 Gigabyte (٤)
Gigabyte 1024 = (TB) 1 Terabyte (٥)

Memory Units...



There are more:

- Petabyte (PB) 10^{15} (Facebook Servers 10+ PBs)
- Exabyte (EB) 10^{18} (amount of data transfer over the internet)
- Zettabyte (ZB) 10^{21} (whole data in the world just a few ZB)
- Yottabyte (YB) 10^{24} (Huge, No practical use)

سرعة وحدة المعالجة المركزية (CPU)

وهو مكون من (Transistor) الذي يتكون من مئات الملايين ، وهو مفتاح كهربائي يسمح بمرور الكهرباء أو يمنع مرور الكهرباء أي (0)(ON) أو (Off) (1)(Binary) وهي اللغة التي يفهمها الكمبيوتر

تقاس سرعة المعالج بالهيرتز (Hertz (Hz وهو عدد الدورات في الثانية الواحدة مثل 2.60GHz، يحدث في هذه الثانية تشغيل وإيقاف ملايين ال(Transistor) مثال : وحدة (CPU) التي تبلغ سرعتها (3.2 GHz) تنفذ 3.2 مليار دورة في الثانية

الدرس الخامس: النظام السادس عشر في الكمبيوتر (Hexadecimal)

النظام السادس عشر (Hexadecimal) مكون من ١٦ خانة يبدأ من 0 الى 9 ، ومن ١٠ الى ١٥ تمثل بالأحرف الإنجليزية من A الى F

١٥	١٤	١٣	١٢	١١	١٠	٩	٨	٧	٦	٥	٤	٣	٢	١	٠
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

ما الفائدة من (Hexadecimal) هي حل وسط بين فهم لغة الكمبيوتر والمبرمج

تتمثل كل Byte برمز ، هو اختصار لقراءة العدد العشري بشكل مختصر (والنظام الثنائي يحول الى النظام العشري) (11010100=D4)

العدد العشري : ٤٦٩١٠ = Hex 1D5١٦ النظام السادس عشر

Base16	16 ⁰	16 ¹	16 ²	16 ³	16 ⁴
Doubling	1	16	256	4096	65536

لكل Hexadecimal رمز اختصار (Prefix) لكل لغة برمجة

Prefix:

Technology/Language	Prefix	Example
HTML & CSS	#Code	#FFFFFF
C,C++,Java..etc	0xCode	0xC2A4
XML	&#Code	&#C2A4
Unicode	U+Code	U+C2A4

مثال حول من النظام العشري ٤٦٩١٠ (النظام العشري (Decimal)) الى Hexadecimal

Hex	الناتج	الكسور * ١٦	العدد الصحيح	الناتج	الرقم / ١٦
5	٥	١٦ * ٠,٣١٢٥	٢٩	٢٩,٣١٢٥	١٦ / ٤٦٩
D	١٣	١٦ * ٠,٨١٢٥	١	١,٨١٢٥	١٦ / ٢٩
1	١	١٦ / ١

يقراً Hex من اليسار = 1D5 (النظام ١٦ (Hex)) 1D5١٦

مثال : حول 1D5 Hex الى النظام العشري (Decimal)

الناتج	تبسيط	Hex * 16
5	5 * 1	5 * 16 ⁰
208	13 * 16	D * 16 ¹
256	1 * 256	1 * 16 ²
469	جمع النتائج	Decimal

مثال : حول 1D5 الى النظام الثنائي (Binary) خطوتين

(١) حول النظام السادس عشر (Hex) الى النظام العشري (Decimal)

(٢) ثم حول نتيجة (Decimal) الى النظام الثنائي (Binary)

مثال : حول النظام الثنائي (Binary) الى نظام السادس عشر (Hex) خطوتين

(١) حول النظام الثنائي (Binary) الى النظام العشري (Decimal)

(٢) ثم حول نتيجة النظام العشري (Decimal) الى نظام السادس عشر (Hex)

(الوسيط دائما يكون العدد لعشري (Decimal)

النتائج (Decimal) = 469 مجموع أرقام

Byte1	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	Base2
Decimal	128	64	32	16	8	4	2	1	Doubling
Byte2	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	Base2
Decimal	32,768	16,384	8,192	4,096	2,048	1,024	512	256	Doubling
	النظام الثنائي (Binary)				111010101				النتائج

الدرس السادس : أجزاء البايت (Byte) ومصطلحاتها

❖ Byte 1 = 8 Bit

❖ أصغر رقم أو قيمة (2⁰) في البت Bit تسمى (LSB) Lowest Significant Bit

❖ أكبر رقم أو قيمة (2⁷) في البت Bit تسمى (MSB) Most Significant Bit

❖ الخانة الواحدة من Byte هي (Bit) وهي مكونة من جزء واحد يخزن فيها

إما ٠ أو ١

❖ خانتين من البت Bit بجانب بعض (Bit2) تسمى Crumb

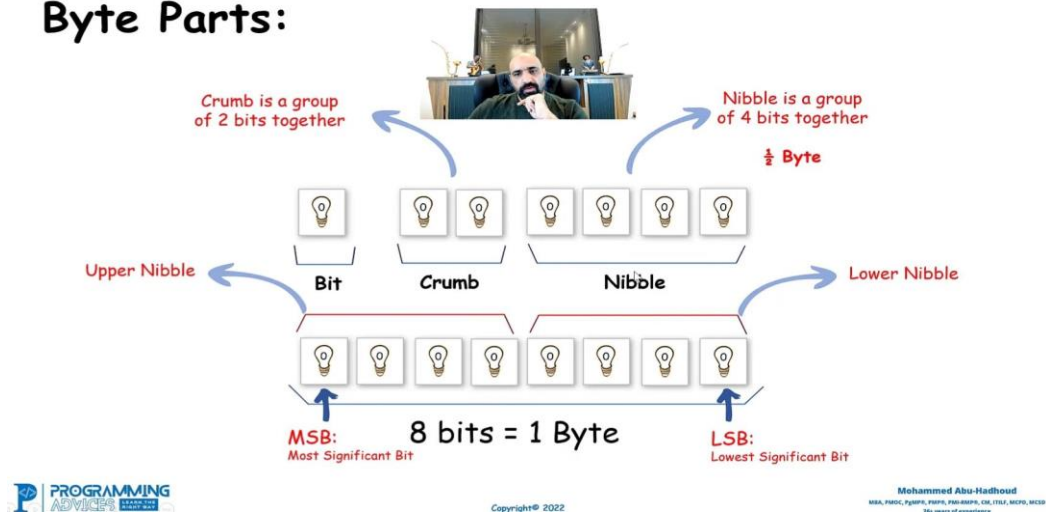
❖ أربع خانات من البت Bit بجانب بعض (Bits4) تسمى Nibble ويسمى

نصف Byte

❖ النصف الأول من البايت Byte يسمى Lower Nibble الأربعة الأولى

❖ النصف الثاني من البايت Byte يسمى Upper Nibble الأربعة الثانية

Byte Parts:



Answer the following questions:

1. What is bit?
2. What is Crumb?
3. What is Nibble?
4. What is Byte?
5. How many bits in a byte?
6. How many crumbs in a byte?
7. How many nibbles in a byte?
8. How many crumbs in a nibble?
9. What is the lower nibble?
10. What is the upper nibble?
11. What is LSB? And why we call it LSB? What it's value?
12. What is MSB? And why we call it MSB? What it's value?



الدرس السابع: النظام السادس عشر الجزء الثاني
التحويل من Hexadecimal الى Binary مباشرة بدون وسيط Decimal
وهي الطريقة التي يستخدمها الكمبيوتر

مثال حول 1D5 الى Binary (لابد من استخدام Nibble)

١٥	١٤	١٣	١٢	١١	١٠	٩	٨	٧	٦	٥	٤	٣	٢	١	٠
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
2^7		2^6		2^5		2^4		2^3		2^2		2^1		2^0	
128		64		32		16		8		4		2		1	

1D5											
1				١٣ = D				5			
0	0	0	1	1	1	0	1	0	1	0	1
000111010101											

تنظر لكل رقم بشكل منفصل وتحوله الى Binary باستخدام Nibble
وليس Decimal بجمع ما يمثله من أرقام، ثم تضم جميع Binary مع بعض

مثال حول 000111010101 Binary الى Hexadecimal

000111010101											
0001				1101				0101			
1				13 = D				5			
				1D5				Hex			

الدرس الثامن : النظام الثماني (Octal System) في الكمبيوتر النظام الثماني مكون من ثمان خانات من 0 الى 7

- ❖ شبيه ب Hexadecimal (يستخدم Nibble أي Bits4)
- ❖ أما النظام الثماني (Octal System) يستخدم Bits3
- ❖ لأن في الماضي كان Byte1 = Bits6 ويقسم الى نصفين أي Bits3 مثل Nibble ،
- ❖ وهو نادر استخدامه الآن ، تجده في بعض الأنظمة القديمة ،
- ❖ لكن الآن يستخدم Hexadecimal لأن كل الحواسيب تستخدم نظام Bits8 الذي ينقسم على Nibble
- ❖ وجد النظام (Hex , Oct) لاختصار قراءة Binary
- لـ Hexadecimal رمز اختصار (Prefix) لكل لغة برمجة

Prefix:

Hexa Prefix:

Technology/Language	Prefix	Example
HTML & CSS	#Code	#FFFFFF
C,C++,Java..etc	0x Code	0x725
XML	&#Code	&#C2A4
Unicode	U+Code	U+C2A4

Octal Prefix:

0oCode → 0o725



مثال حول 469 الى Octal

الرقم / ٨	الناتج	العدد الصحيح	الكسور * ٨	الناتج	Oct
٨/٤٦٩	٥٨,٦٢٥	٥٨	٨* ٠,٦٢٥	٥	5
٨/٥٨	٧,٢٥	٧	٨* ٠,٢٥	٢	2
٨/٧					7
Decimal = 469 ₁₀ Oct = 725 ₈					

مثال حول 725 Octal الى Decimal

Oct * 8	تبسيط	الناتج
5 * 8 ⁰	5 * 1	5
2 * 8 ¹	2 * 8	16
7 * 8 ²	7 * 64	448
Decimal	جمع النتائج	469

مثال حول 725 Octal الى Binary

يوجد طريقتين

(١) الطريقة الطويلة

• حول النظام الثماني (Octal) الى النظام العشري (Decimal)

• ثم حول نتيجة (Decimal) الى النظام الثنائي (Binary)

(٢) الطريقة المختصرة

تنظر لكل رقم بشكل منفصل وتحوله الى Binary وكل رقم يحول الى ثلاث

خانات 3 Bits ثم جمع كل خانة ما تمثله من أرقام، ثم تضم جميع Binary

مع بعض

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

725								
7			2			5		
1	1	1	0	1	0	1	0	1

حول من Binary الى Octal

طريقتين

(١) الطريقة الطويلة

- حول النظام الثنائي (Binary) الى النظام العشري (Decimal)
 - ثم حول نتيجة النظام العشري (Decimal) الى نظام الثماني (Octal)
- (الوسيط دائما يكون العدد لعشري Decimal)
- (٢) الطريقة المختصرة نقسم كل ثلاث Bits الى مجموعة

000111010101								
1	1	1	0	1	0	1	0	1
7			2			5		
725								

الدرس التاسع : آلة حاسبة المبرمجين

من خلال ويندوز الحاسبة ثم مبرج

الدرس العاشر : الشبكات

شبكات الاتصال (Network) هي مجموعة من أجهزة الكمبيوتر أو أي جهاز إلكتروني المتصلة معا ضمن مجموعة أو شبكة واحدة

نحتاج شبكة الاتصال لإرسال البيانات من جهاز لجهاز ، وسهولة تواصل وتبادل المعلومات أو البيانات أو الأوامر

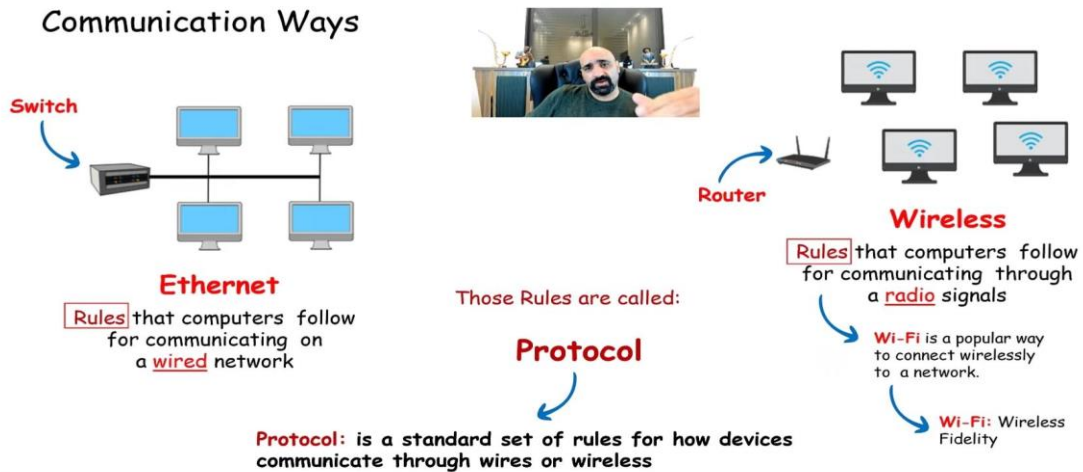
أنواع شبكة الاتصال

الشبكة المحلية 'LAN' (Local Area Network)

الشبكة العريضة 'WAN' (Wide Area Network)



Communication Ways



الشبكة المحلية LAN : مجموعة من الأجهزة المتصلة مع بعض

وميزتها أن تكون الأجهزة قريبة من بعضها على شبكة واحدة

كيف تتواصل الأجهزة مع بعض LAN إما

❖ سلكي 'Wired' باستخدام الأسلاك أو الكابلات ، تسمى إيثرنت **Ethernet**

❖ لاسلكي 'Wireless' بدون أسلاك باستخدام Wi-Fi (Wireless Fidelity)

الشبكة العريضة WAN هي مجموعة من الشبكات المحلية

تستخدم للمسافات البعيدة التي قد تتكون من ملايين الأجهزة

الدرس الحادي عشر : لغات البرمجة - الجزء الأول

لغة البرمجة : هي مجرد أداة لمساعدتك في كتابة الأوامر الى الكمبيوتر لأداء مهمة معينة

الأوامر هي (Code) هو مجموعة من الأوامر يستخدمها المبرمج داخل لغة برمجة

Hello World! لكتابة

قبل ظهور لغات البرمجة كانت تكتب الأوامر بلغة الآلة (الكمبيوتر) **Machin**

Language وهي لغة برمجة مكونة من Hexadecimal أو Binary يفهمها الكمبيوتر (CPU) بشكل مباشر وهي أسرع لغة ، ولأنها غير مفهومة لدى الإنسان اخترعوا لغة **Assembly Language** أسهل من لغة الآلة (الكمبيوتر) بقليل

وهذه اللغات التي لا يفهمها البشر تسمى **Low Level Languages** ،

ولأنها صعبة لدى البشر اخترعوا لغات البرمجة مثل

C++ , Basic (High Level Language) التي يفهمها البشر

High Level Language ولأنها لغة لا يفهمها الكمبيوتر يحتاج وسيط مترجم
Translator Software وظيفته الوحيدة الترجمة من High... الى Machine إما

Interpreter ❖

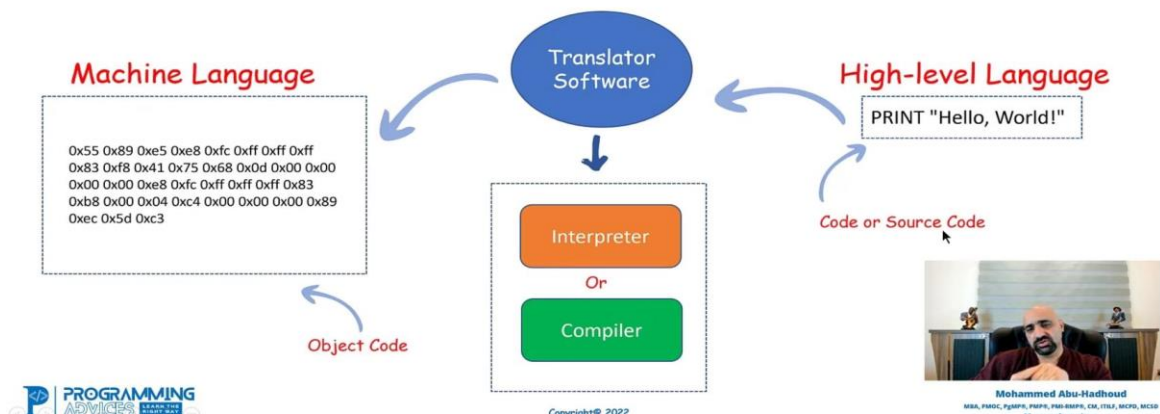
Compiler ❖

هما Translator Software يقومان بترجمة High Level Language اللغة
التي يفهمها البشر الى لغة الكمبيوتر Machine Language

High Level Language يسمى Code or Source Code

Machine Language يسمى Object Code

You need a Translator!



درجات لغات البرمجة التي يفهمها الهاردوير هي Binary

(١) Lowest Level Machine Language أو Object Code

- Hexadecimal

- Binary Instructions

(٢) Low Level Languages

- Assembly

(٣) Middle Level Languages

- C , C++

(٤) High Level Languages

- Java , C# , VB , etc

(٥) Scripting \ Interpreted Languages

- Python, JavaScript

❖ كلما كانت لغات البرمجة أقرب الى الهاردوير كانت لغات البرمجة Low

والعكس كانت High

❖ كلما قربت من الهاردوير كانت اللغة صعب تعلمها Low والعكس High

❖ كلما كانت اللغة أقرب الى الهاردوير كانت أسرع High والعكس Low



الدرس الثاني عشر : لغات البرمجة الجزء الثاني - الفرق بين Compiler و Interpreter

Interpreter ❖

Compiler ❖

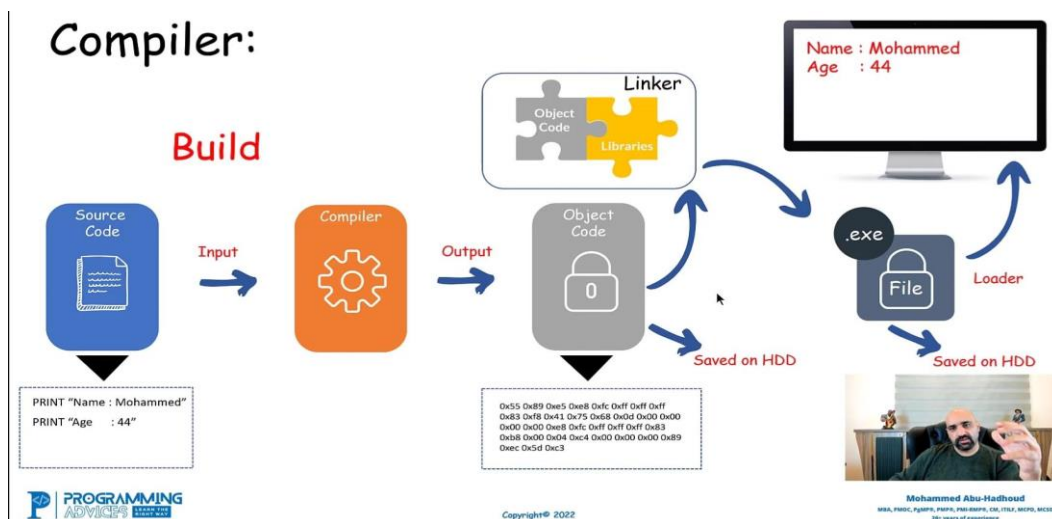
هما Translator Software يقومان بترجمة High Level Language اللغة التي يفهمها البشر الى لغة الكمبيوتر Machine Language

Compiler Language

برنامج مكتوب ب High Level Language (Source Code) إذا أرت تشغيله في Compiler يطلق على أمر التشغيل أمر Build للبرنامج يعني أنك أمرت Compiler أن يترجم Source Code الى (Machine Language) Object Code ويشغل معه Linker أي يربط بين أوامرك والمكتبة الخاصة بلغة البرمجة مثل (print) وينتج عن ذلك ملف يحفظ فيه Compiler اسمه File.exe هو ناتج عن Source Code و Object Code مع المكتبة

إذا شغلت أو فتحت File.exe يعطي لنا أمر Loader هو تحميل الملف على ذاكرة الجهاز وينتج لنا الأوامر المخزنة في الملف

File.exe هو ملف لا يمكن تعديله ، ويمكن نقله الى جهاز آخر ، ويخزن في الهارد ديسك ويخزن أيضا Object Code



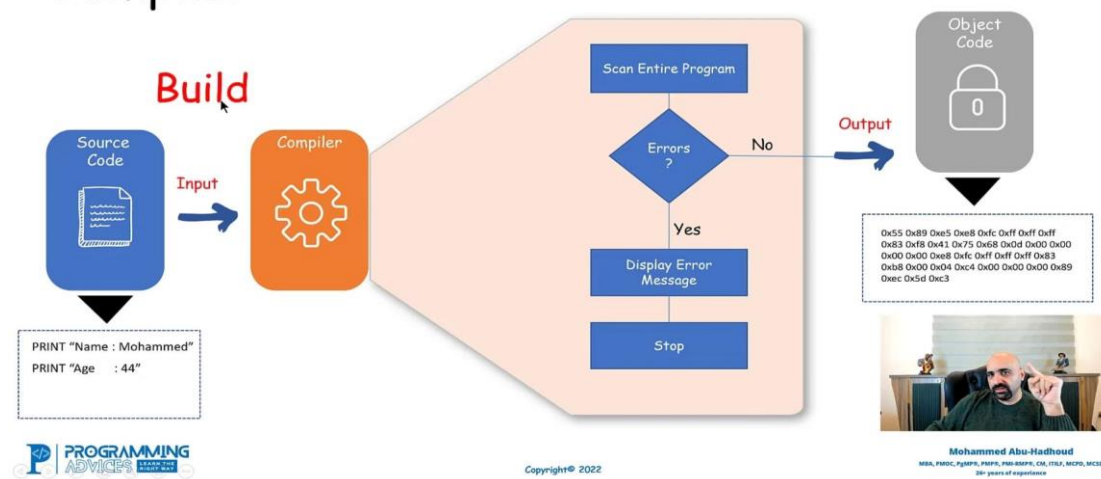
كيف يعمل Compiler

بعد كتابة Source Code نعطي أمر التشغيل Build ثم يترجم Compiler الى Machine Language

طريقة عمل Compiler

- يقرأ كل البرنامج مرة واحدة
- بعد القراءة يبحث عن الأخطاء مثل (خطأ في قواعد كتابة اللغة 'brint')
- إذا وجد خطأ يتوقف البرنامج ويظهر الخطأ الذي وجده
- إذا لم يجد أخطاء ينتج لنا Object Code ويشغل معه Linker أي يربط بين أوامرك والمكتبة الخاصة بلغة البرمجة مثل (print)
- وينتج عن ذلك ملف اسمه File.exe

Compiler:



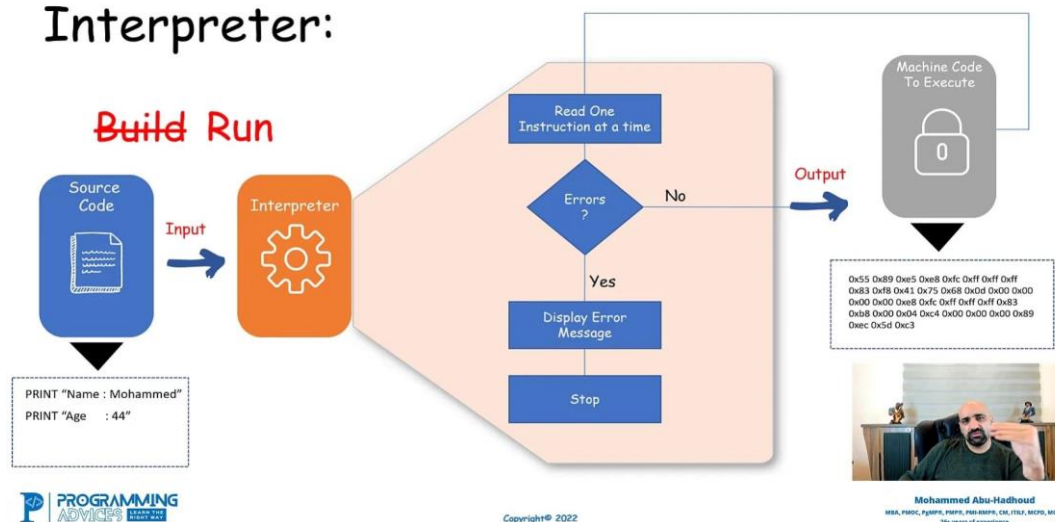
Interpreter Language

طريقة عمله مثل Compiler إلا أنه لا يوجد Build أي بناء فقط يوجد أمر تشغيل Run

طريقة عمل Interpreter

- يقرأ البرنامج سطر سطر أمر
- بعد القراءة يبحث عن خطأ في أمر مثل (خطأ في قواعد كتابة اللغة 'brint')
- إذا وجد خطأ يتوقف البرنامج ويظهر الخطأ الذي وجده
- إذا لم يجد خطأ في السطر ينفذ السطر ثم يقرأ السطر التالي
- لهذا لا يحفظ في الهارد ديسك Object Code و File.exe

Interpreter:



من الأسرع Interpreter أو Compiler؟

Compiler أسرع بكثير من Interpreter

الفرق بين Compiler و Interpreter

Compiler	Interpreter
يقرأ كل الأوامر دفعة واحدة	يقرأ أمر واحد في كل مرة
تنفيذ العملية أسرع بكثير	تنفيذ العملية أبطأ بكثير
يخزن Object Code في HDD	لا يخزن Object Code في HDD
يتم إنشاء ملف File.exe	لا يتم إنشاء ملف File.exe
مثال للغة البرمجة C , C++	مثال للغة البرمجة Python, JavaScript

ما هو Assembler ؟

هو المترجم للغة Assembly مثل مترجم Machine code

الدرس الثالث عشر : لغات البرمجة - الجزء الثالث
السؤال الصحيح للسؤال عن أفضل لغة برمجة هو
ما هي أفضل لغة برمجة في (كذا ، التطبيقات ، الويب)
لكل لغة برمجة نقاط قوة

تشارك جميع لغات البرمجة في نفس المفاهيم ، والفرق الوحيد الاختلاف في قواعد
كتابة اللغة و بعض الأدوات

البرمجة هي أسلوب تفكير وتحليل منطقي للمشكلة لإيجاد أفضل الحلول

البرمجة ...

عملية لإنشاء مجموعة من التعليمات التي تخبر الكمبيوتر بكيفية تنفيذ مهمة ما.

كل هذا يتوقف على:

منطق التحليلات التفكير القدرة على البحث

بغض النظر عن أي برمجة لغة!

موقعي على YouTube

Programming is...

A process of creating a set of instructions that tell a computer how to perform a task.

It all depends on:

Logic Analysis Thinking Ability to find solutions

Regardless of any programming language!

YouTube ProgrammingTutorials Mohammed Abu-Hadoud

الدرس الرابع عشر: لماذا يجب أن تبدأ بتعلم C++
لأن هذه اللغة تأسسك بشكل قوي وتجبرك بتعلم مفاهيم البرمجة كلها
إذا تعلمت لغة C++ تستطيع تعلم أي لغة برمجة في أقل من شهر

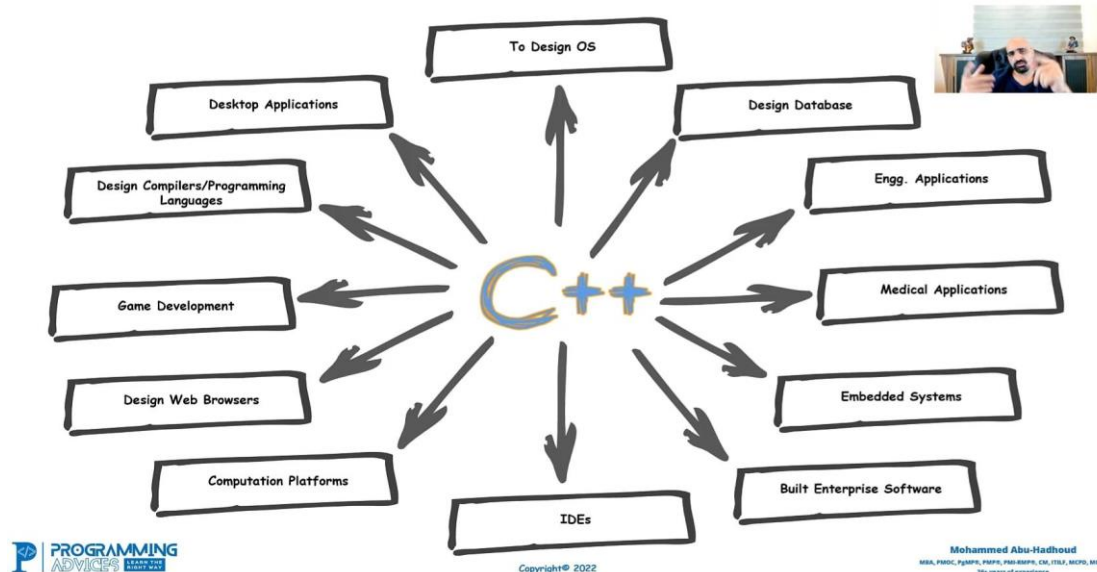
ما هي لغة C++ ؟

هي لغة متعددة المنصات ، يمكن إنشاء تطبيقات عالية الأداء على أنظمة مختلفة

- ❖ ممتعة وسهلة التعلم
- ❖ هي أم اللغات، لأن لغات البرمجة الأخرى مبنية بشكل مباشر أو غير مباشر على C++
- ❖ تدعم البرمجة الهيكلية (Procedural) & والبرمجة كائنية التوجيه OOP
- ❖ ستجبرك على تعلم مفاهيم البرمجة الأساسية
- ❖ تستطيع تعلم أي لغة برمجة في أقل من شهر
- ❖ تتمتع C, C++ بالتحكم الكامل في موارد الكمبيوتر (الهارد وير) مثل الذاكرة ، ولا توجد بها إدارة تلقائية للذاكرة

ما الذي تستطيع عمله بلغة C++ ؟

١. أنظمة تشغيل	٢. تطبيقات مثل الفوتوشوب	٣. لغات برمجة
٤. قواعد بيانات	٥. مخططات هندسية	٦. ألعاب ضخمة
٧. برامج الأجهزة الطبية	٨. متصفح انترنت	٩. الأنظمة المضمنة مثل تسلا
١٠. البرامج الحسابية	١١. برامج ضخمة	١٢. IDEs فيجول استديو



المكتبات

عندما نحتاج الى حسابات رياضية عالية المستوى ، يصبح الأداء والسرعة مهمين لذا فإن معظم المكتبات تستخدم C++ كلغة برمجة أساسية ، تستخدم معظم مكتبات لغات High-level لغة C++ كخلفية لها تعد لغة C++ أسرع من معظم لغات البرمجة الأخرى كما أنه يدعم تعدد مؤشرات الترابط مع التزامن

مميزات C++

- أداء
- سرعة
- قابلة للتكيف متعددة المنصات ، يمكن إنشاء تطبيقات عالية الأداء على أنظمة مختلفة
- تعدد العمليات الحسابية (معقدة ، كثيرة)
- عالية المستوى مع التزامن

مقارنة بين C++ و C

أوجه التشابه

C	C++
نفس syntax قواعد الكتابة أو بناء الجملة وكتابة الكود	نفس syntax قواعد الكتابة أو بناء الجملة وكتابة الكود
تستخدم لغة ترجمة Compiler	تستخدم لغة ترجمة Compiler
كلاهما لغات متوسطة المستوى ، قريبة من الهارد وير	كلاهما لغات متوسطة المستوى ، قريبة من الهارد وير
التحكم الكامل في موارد (الهارد وير) الكمبيوتر مثل الذاكرة	التحكم الكامل في موارد (الهارد وير) الكمبيوتر مثل الذاكرة

أوجه الاختلاف

C	C++
أنشأت بين 1969 الى 1973	أنشأت عام 1979
لغة C جزء من C++	لغة C++ أشمل من C
برمجة هيكلية فقط (Procedural)	برمجة هيكلية و OOP

مقارنة بين C++ و Java

Java	C++
برمجة OOP فقط	برمجة هيكلية و OOP

الدرس الخامس عشر : العمليات في الكمبيوتر

- ❖ عمليات حسابية (Mathematical) مثل [+ , - , * , /]
- ❖ عمليات مقارنة (Relational) مثل [< , > , = , <= , >=]
- ❖ العمليات المنطقية (Logical) مثل [And , or , not]

١. عمليات حسابية (Mathematical)

Mathematical Operators



Operator	Computer Symbol	Example	
		Operation	Results
Addition	+	3.0 + 4.2	7.2
Subtraction	-	8.5 - 4.0	4.5
Multiplication	*	5 * 5	25
Integer Division	/	10 / 5	2
Modulo Division	Mod	9 Mod 4	1
Power	^	3^2	9

٢. عمليات مقارنة (Relational)

Relational Operators



Operator	Computer Symbol	Example	
		Operation	Results
Equal to	=	5 = 7	False
Less than	<	5 < 7	True
Grater than	>	5 > 7	False
Less than or equal to	<=	5 <= 7	True
Grater than or equal to	>=	5 >= 7	False
Not equal to	<>	5 <> 7	True

٣. العمليات المنطقية (Logical)

- And لابد من تحقق جميع الشروط لتصبح النتيجة True (1)
- Or لابد من تحقق أحد الشروط لتصبح النتيجة True (1)
- Not يعكس النتيجة إذا كانت True (1) تكون النتيجة False (0)

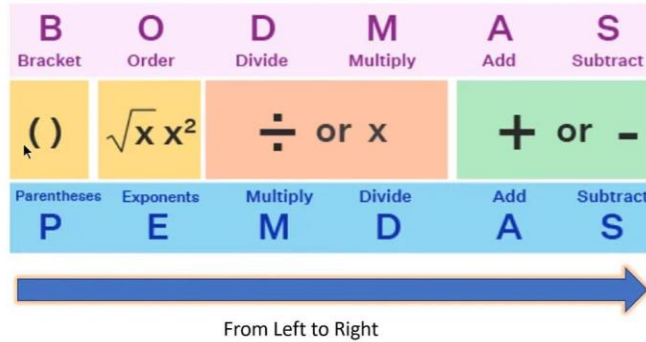
الدرس السادس عشر : أولويات الحساب (Calculation Priorities)

في المعادلات الحسابية هناك أولويات يجب البدء بها هي :

يجب البدء من اليسار الى اليمين ، إذا تساوت درجة الأولويات

- (١) الأقواس () كأنها معادلة منفصلة ، إذا وجد أقواس داخل قوس تأخذ أصغر قوس في المعادلة
- (٢) الجذر التربيعي، والأسس ^
- (٣) القسمة /، والضرب *
- (٤) الجمع +، والناقص -

Order of Operations



الدرس السابع عشر : المتغيرات (Variables) & والثوابت (Constants)

المتغيرات (Variables)

المتغيرات عبارة عن حاويات لتخزين البيانات (تستخدم المتغيرات لتخزين البيانات) يكون تخزين المعلومات مفيداً عندما تريد إعادة استخدام هذه المعلومات عدة مرات أو استخدامها في أوقات مختلفة أثناء تشغيل أحد البرامج

تستطيع تغيير قيمة المتغير أثناء البرنامج

أنواع تخزين بيانات المتغيرات

(١) أرقام Numbers

• (10 , 22) Integers

• (4.5 , 8.9) Floats

(٢) النصوص ("Saeed Omar") Strings

(٣) القيم المنطقية Boolean

• True صح

• False خطأ

كيف يتم حفظ المتغيرات (Variables) في RAM يتكون من ملايين الخلايا Cell

مثال

Int Age = 23

يحجز لها مكان في الذاكرة RAM تسمى Memory Cell خلية في الذاكرة

تحديد جزء من الذاكرة للبيانات ، مثل عمر الإنسان لا يتجاوز ١٥٠ سنة لحجز ١٥٠ بت من الذاكرة بدلا من أربعة مليار

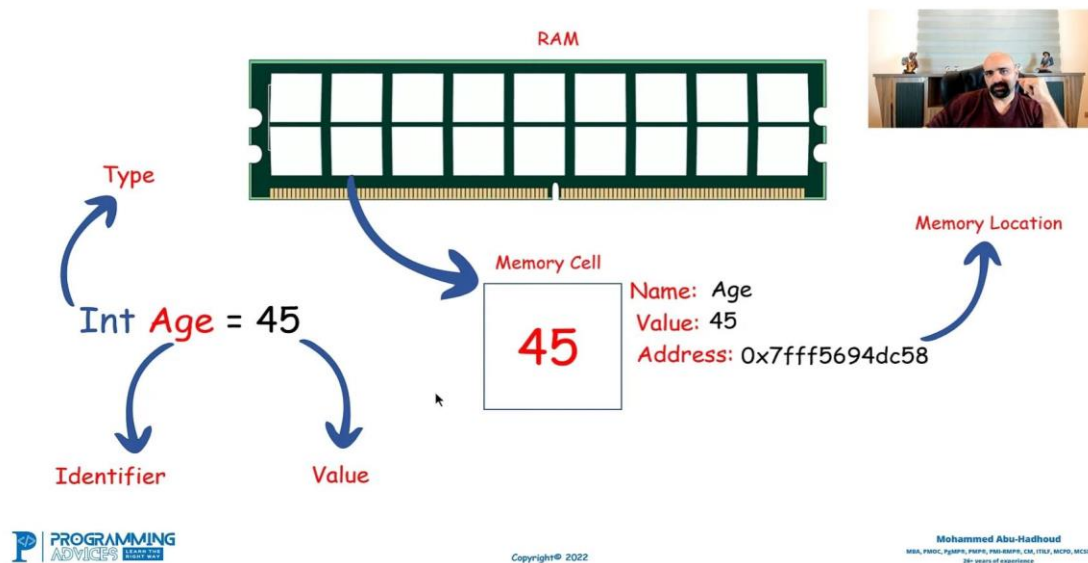
Memory Cell خلية في الذاكرة			
23	Name: الاسم:	Age	Identifier اسم المتغير Variable
	Value: القيمة:	23	Value القيمة
	Address: العنوان:	يكتب ب HEX	Type نوع Int

لكل Memory Cell خلية في الذاكرة لها

Name: الاسم:
Value: القيمة:
Address: العنوان:

Address يدل على مكان التخزين

تستطيع تغيير قيمة Variables في البرنامج



الثوابت (Constants)

عبارة عن حاويات لتخزين قيم البيانات، لكنها للقراءة فقط (Const)

