

جامعة دمشق
كلية الهندسة المدنية

مادة البرمجة
السنة الثالثة

29/3/2023

Wael Darwich

1

Content

- Programming for Civil Engineers
 - Rational Thinking
- Programming Languages
 - MS Visual Basic
- Examples
- Syllabus
- Memory and Variables

2

To Program or Not to Program

- Programming for Civil Engineers
- Software Developers
- Simulation, Design and Analysis Tools
 - FEA
- Software End User
- Automation
- Logic and Rational Thinking

29/3/2023

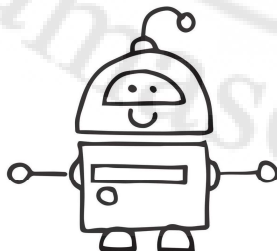
Programming for Civil Engineers

3

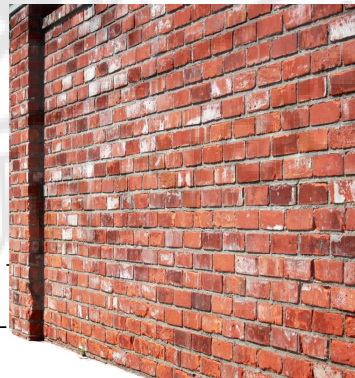
3

Rational Thinking

- Smart Computer/Software
- GIGO: Garbage In Garbage Out



X →



29/3/2023

Programming for Civil Engineers

4

4

Rational Thinking

- Smart Computer/Software
- GIGO: Garbage In Garbage Out
- Algorithm
 - Set of Instructions
 - Inputs
 - Outputs
- Numerical issues
 - Point Inside Circle

Input



Process



Output

29/3/2023

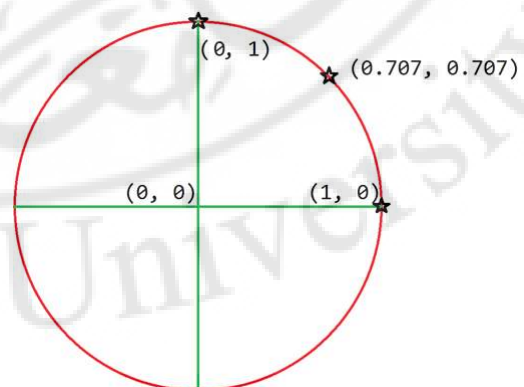
Programming for Civil Engineers

5

5

Rational Thinking

- Smart Computer/Software
- GIGO: Garbage In Garbage Out
- Algorithm
 - Set of Instructions
 - Inputs
 - Outputs
- Numerical issues
 - Point Inside Circle



29/3/2023

Programming for Civil Engineers

6

6

Programming Languages

- Fortran Formula Translation, 1957
- BASIC Beginner's All-purpose Symbolic Instruction Code, 1964
- C/C++ 1972/1985
- MS Visual Basic 1991
- Python 1991
- Java 1995

29/3/2023

Programming for Civil Engineers

7

7

Hello World

program HelloWorld	#include <stdio.h>
print *, "Hello World"	#include <stdlib.h>
end program HelloWorld	
10 PRINT "Hello World!"	int main(void)
#include <iostream>	cout << "Hello World!" << endl;
int main()	return 0;
{	
std::cout << "Hello World!" << std::endl;	
return 0;	
}	

8

8

Hello World

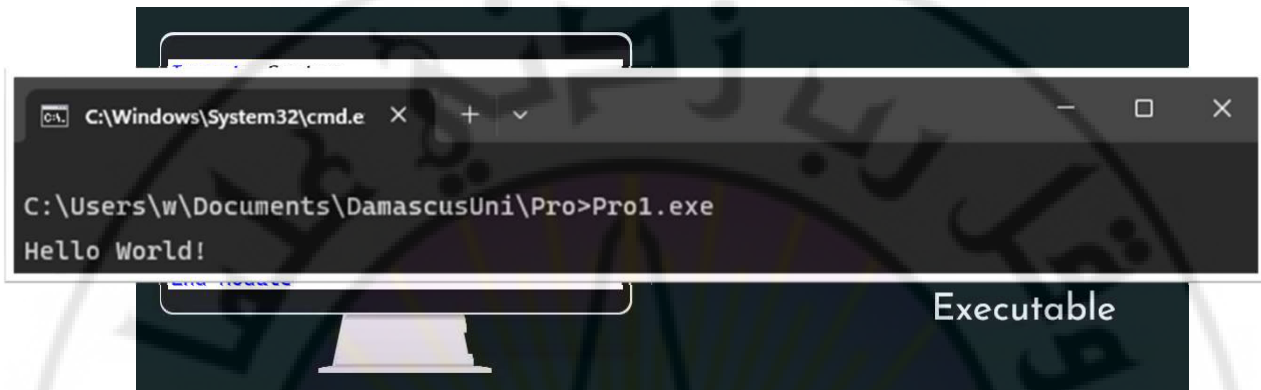
```
print("Hello World!")
```

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Visual Basic

```
Imports System
Module Program
    Sub Main(args As String())
        ' Hello World!
        Console.WriteLine("Hello World!")
    End Sub
End Module
```

Visual Basic



29/3/2023

Programming for Civil Engineers

11

11

Programming Languages

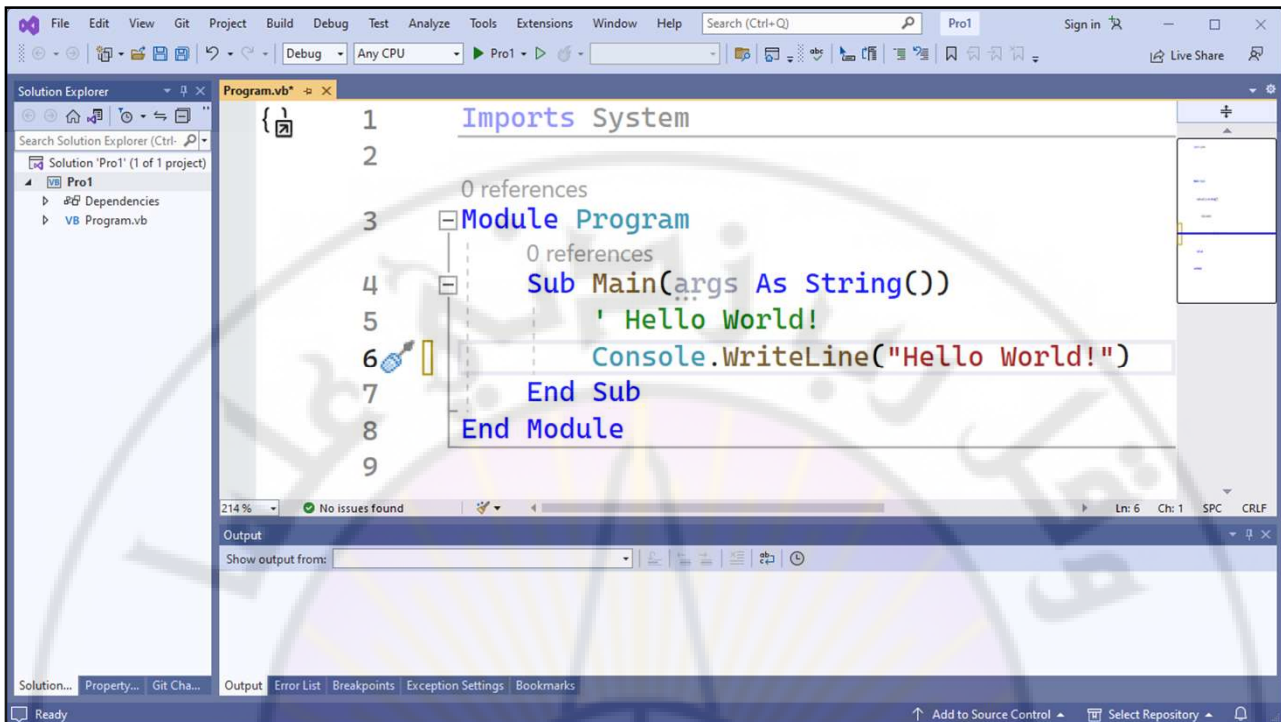
- Source Code
- Keywords
- Syntax
- Compiler vs. Interpreter
- IDE Integrated Development Environment
 - Text Editor
 - Compiler
 - Debugger

29/3/2023

Programming for Civil Engineers

12

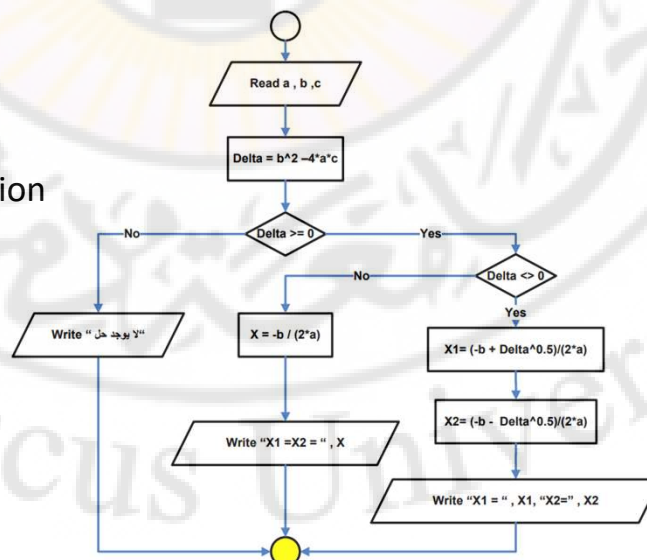
12



13

Examples

- Second Order Equation
- Version 2.0
 - Imaginary Solution
- Crash!
 - First Order



معلوماتية سنة أولى - د عبد السلام زيدان

14

Examples

• حساب كلفة فاتورة الكهرباء

التعرفة:

من ١ إلى ٦٠٠ بسعر ٢ ل.س، من ٦٠١ إلى ١٠٠٠ بسعر ٦ ل.س، من ١٠٠١ إلى ١٥٠٠ بسعر ٢٠ ل.س، من ١٥٠١ إلى ٢٥٠٠ بسعر ٩٠ ل.س، من ٢٥٠١ فما فوق بسعر ١٥٠ ل.س لكل ك.وس

عزيزي المشترك:

عزيزي المشترك يرجى عدم احضار اي مبلغ فوق المليون ليرة سورية وتسديد قيمة الفاتورة عن طريق (نافذة الدفع الالكتروني) او عن طريق شيك من اي مصرف مصدق اصولا او عن طريق المصرف التجاري في حساب الشركة العامة لكهرباء محافظة دمشق رقم الحساب/ ١٠٥١٠١٤٣٩٠٠١ / ومراجعتنا بعد التسديد للحصول على الفاتورة مباشرة نرجو المبادرة الى تسديد الفواتير المترتبة عليكم تجنباً من سحب العداد في حال التأخر لمدة اربعة اشهر تطبيقاً لنظام الاستثمار الجديد .

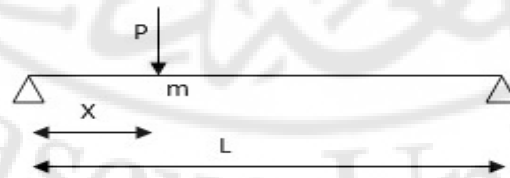
29/3/2023

Programming for Civil Engineers

15

Examples

- Calculate reactions and bending moment

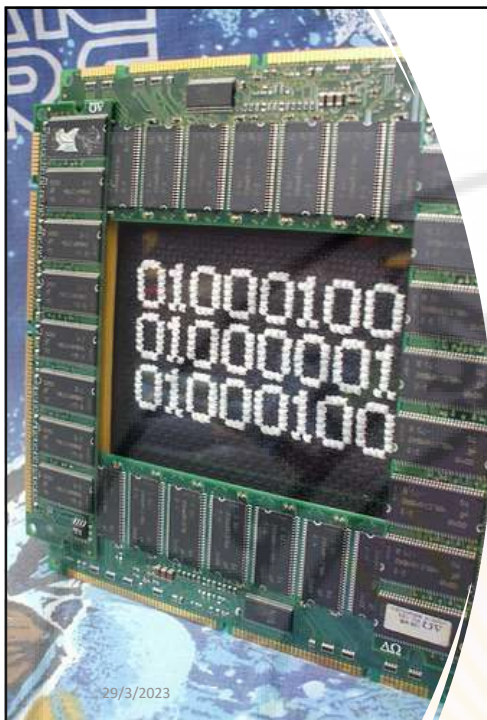


29/3/2023

Programming for Civil Engineers

16

16



Bit vs Byte

- Bit or binary digit
- 2 Bit
- 3 Bit

0/1

0/1 0/1

0/1 0/1 0/1

2^3 values = 8

0:	0	0	0
1:	0	0	1
2:	0	1	0
3:	0	1	1
4:	1	0	0
5:	1	0	1
6:	1	1	0
7:	1	1	1

29/3/2023

Programming for Civil Engineers

19

19

Memory

- Input 'Hello' to memory
- Hello = H + e + l + l + o

29/3/2023

Programming for Civil Engineers

20

20

ASCII

- The **American Standard Code for Information Interchange**
- $2^8 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$



29/3/2023

Programming for Civil Engineers

21

21

DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol	DEC	Symbol
0	NUL	32	!	64	@	96	~	128	€	160	À	192	À	224	à		
1	SOH	33	"	65	A	97	a	129	ı	161	Á	193	Á	225	á		
2	STX	34	#	66	B	98	b	130	ˆ	162	Â	194	Â	226	â		
3	ETX	35	\$	67	C	99	c	131	ç	163	Ã	195	Ã	227	ã		
4	EOT	36	%	68	D	100	d	132	˘	164	Ä	196	Ä	228	ä		
5	ENQ	37	&	69	E	101	e	133	…	165	Å	197	Å	229	å		
6	ACK	38	'	70	F	102	f	134	†	166	Æ	198	Æ	230	æ		
7	BEL	39	(	71	G	103	g	135	‡	167	Ç	199	Ç	231	ç		
8	BS	40	)	72	H	104	h	136	£	168	Ɔ	200	Ɔ	232	ø		
9	HT	41	*	73	I	105	i	137	‰	169	©	201	©	233	é		
10	LF	42	+	74	J	106	j	138	Š	170	ª	202	ª	234	ê		
11	VT	43	,	75	K	107	k	139	ƒ	171	«	203	«	235	ë		
12	FF	44	-	76	L	108	l	140	œ	172	¬	204	¬	236	ì		
13	SO	45	.	77	M	109	m	141	ˆ	173	˜	205	˜	237	í		
14	CR	46	:	78	N	110	n	142	Ž	174	˘	206	˘	238	î		
15	SI	47	;	79	O	111	o	143	ž	175	ı	207	ı	239	ï		
16	DLE	48	<	80	P	112	p	144	˘	176	˘	208	˘	240	ó		
17	DC1	49	=	81	Q	113	q	145	˘	177	˘	209	˘	241	ô		
18	DC2	50	>	82	R	114	r	146	˘	178	˘	210	˘	242	õ		
19	DC3	51	?	83	S	115	s	147	˘	179	˘	211	˘	243	ö		
20	DC4	52	@	84	T	116	t	148	˘	180	˘	212	˘	244	ø		
21	NAK	53	A	85	U	117	u	149	˘	181	˘	213	˘	245	ı		
22	SYN	54	B	86	V	118	v	150	˘	182	˘	214	˘	246	ı		
23	ETB	55	C	87	W	119	w	151	˘	183	˘	215	˘	247	ı		
24	CAN	56	D	88	X	120	x	152	˘	184	˘	216	˘	248	ı		
25	EM	57	E	89	Y	121	y	153	™	185	˘	217	˘	249	ı		
26	SUB	58	F	90	Z	122	z	154	š	186	˘	218	˘	250	ı		
27	ESC	59	G	91	[	123	{	155	˘	187	˘	219	˘	251	ı		
28	FS	60	H	92	\	124		156	œ	188	¼	220	¼	252	ı		
29	GS	61	I	93	]	125	}	157	˘	189	½	221	½	253	ı		
30	RS	62	J	94	^	126	~	158	˘	190	¾	222	¾	254	ı		
31	US	63	K	95	_	127		159	˘	191	˘	223	˘	255	ı		

29/3/2023

Programming for Civil Engineers

22

22

Memory

- Input 'Hello' to memory
- Hello = H + e + l + l + o
- 01001000 01100101 01101100 01101100 01101111

29/3/2023

Programming for Civil Engineers

23

23

Study Material

- Download material from:
 - <http://damascusuniversity.edu.sy/civil/index.php>
 - <https://pro.el.sy/>
- Contact:
 - w@el.sy



29/3/2023

Programming for Civil Engineers

24

24

جامعة دمشق
كلية الهندسة المدنية

مادة البرمجة
السنة الثالثة

5/4/2023

Wael Darwich

1

Hello World! Visual Basic

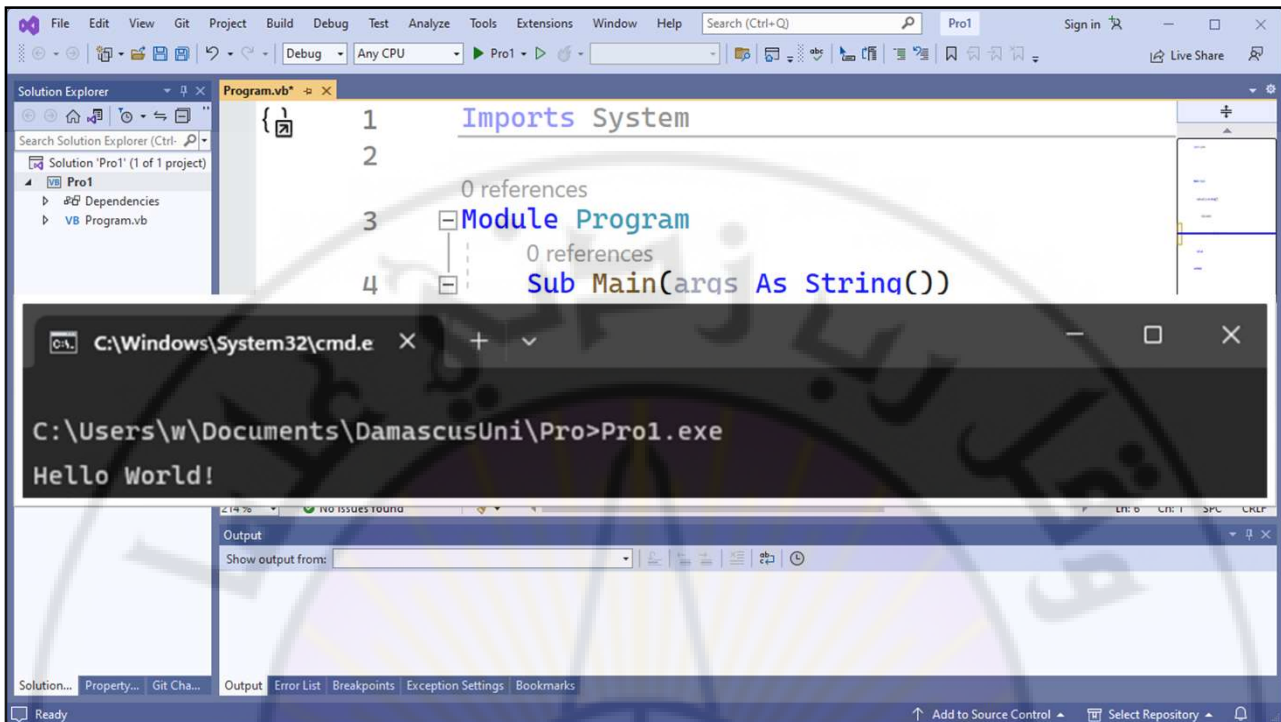
```
Imports System
Module Program
    Sub Main(args As String())
        ' Hello World! Comment with apostrophe
        Console.WriteLine("Hello World!") ' Write line
    End Sub
End Module
```

5/4/2023

Programming for Civil Engineers

2

2



3

Content

- Memory and Variables
- Variables
 - Type
 - Name
 - X Æ A-12
 - X Æ A-XII
 - Size
- Assignment
- Operators
 - Precedence of Operators
- Subroutine
 - Example

4

Example

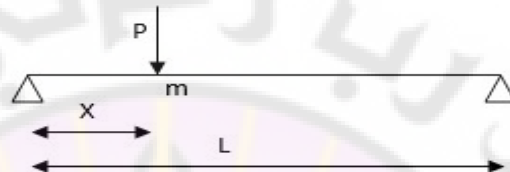
- Calculate reactions and bending moment

- Input:

- L, x, P (real numbers)
- S_L, S_R

- Output:

- R_L, R_R, m



5/4/2023

Programming for Civil Engineers

5

5

Variables

- Reserved locations in memory

- Names:

- Including a-z, A-Z, 0-9, _ (underscore)
- Starting a-z, A-Z, _
- No space `section_radius`
- CamelCase: `SectionRadius`
- Case insensitive
 - $A = 1, a = 2$
 - Same variable
- Meaningful name: `length, width` instead of `myVar`

5/4/2023

Programming for Civil Engineers

6

6

Variable Types

- Boolean: True / False
- Numbers
 - Integers, e.g., counter
 - Decimals
- Strings "Text"
- ...

5/4/2023

Programming for Civil Engineers

7

7

Booleans in Memory



- 1 Byte = 8 Bit
- Could be single byte, or more
 - Packing

5/4/2023

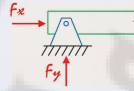
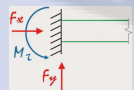
Programming for Civil Engineers

8

8

DoF Booleans

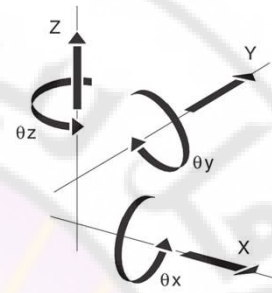
- 2D translation: 2 Booleans
- 3D translation: 3 Booleans
- 3D translation + rotation: 6 Booleans

	Tx	Ty	Tz	Rx	Ry	Rz
Free						
	✓	✓				
	✓	✓				✓

5/4/2023

Programming for Civil Engineers

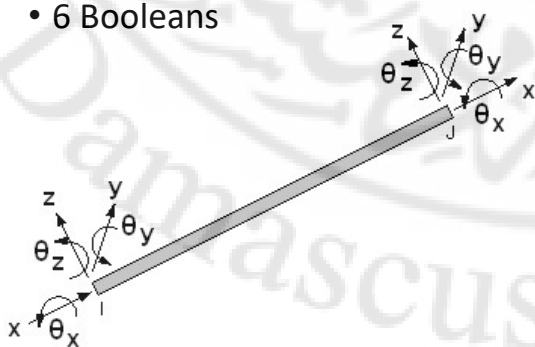
9



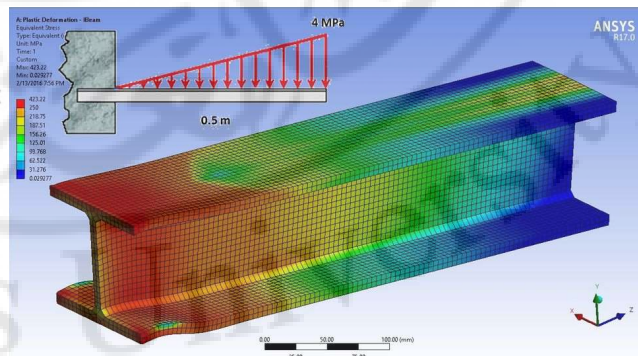
9

DoF Booleans

- 6 Booleans



2 Nodes, 12 Byte



200k Nodes, 1.2 M Byte

5/4/2023

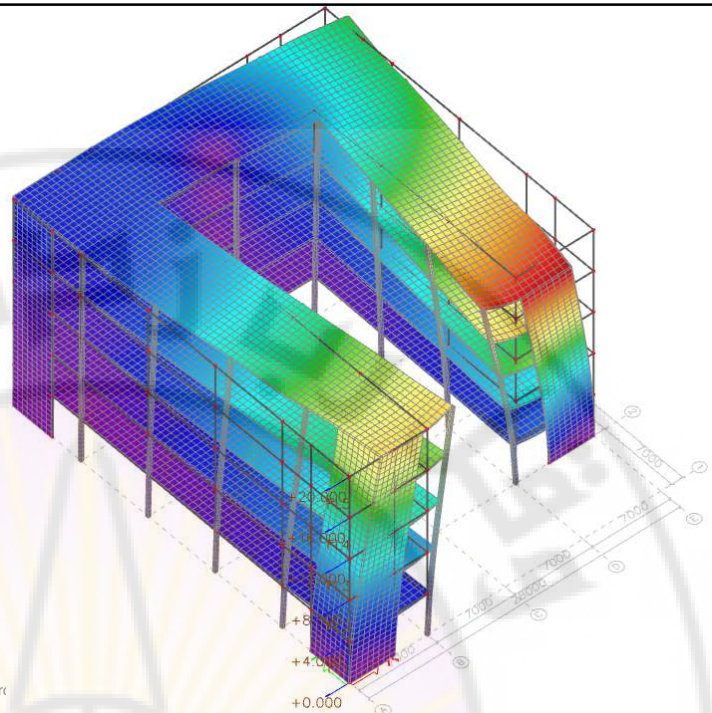
Programming for Civil Engineers

10

10

DoF Booleans

- Geometry
- Support and load conditions
- Material
- Calculate stiffness matrix
- Reduce required memory
 - Preserve information



5/4/2023

Prc

11

Number of Combinations

- 3D translation + rotation: 6 Booleans per point
- Array of 6 Booleans
- (True / False)⁶
- $2^6 = 64$

5/4/2023

Programming for Civil Engineers

12

12

Number of Combinations

0	000000	10	001010	20	010100	30	011110	40	101000	50	110010	60	111100
1	000001	11	001011	21	010101	31	011111	41	101001	51	110011	61	111101
2	000010	12	001100	22	010110	32	100000	42	101010	52	110100	62	111110
3	000011	13	001101	23	010111	33	100001	43	101011	53	110101	63	111111
4	000100	14	001110	24	011000	34	100010	44	101100	54	110110		
5	000101	15	001111	25	011001	35	100011	45	101101	55	110111		
6	000110	16	010000	26	011010	36	100100	46	101110	56	111000		
7	000111	17	010001	27	011011	37	100101	47	101111	57	111001		
8	001000	18	010010	28	011100	38	100110	48	110000	58	111010		
9	001001	19	010011	29	011101	39	100111	49	110001	59	111011		

5/4/2023

Programming for Civil Engineers

13

13

Number of Combinations

- 3D translation + rotation: 6 Booleans per node
- Array of 6 Booleans
- (True / False)⁶
- $2^6 = 64$
- Less than a byte ($2^8 = 256$)
- 1 Byte per node (0-255)
- Less memory
- Same data

5/4/2023

Programming for Civil Engineers

14

14

Char

- 2 Bytes Unicode
- Single character

5/4/2023

Programming for Civil Engineers

15

15

Integers in Memory



- 1 Byte variable
 - Unsigned Integer 0 to 255 ($2^8=256$)
1111 1111 Bin = 255 Dec
 - Signed integer -128 to 127 ($2^7=128$)
111 1111 Bin = 127 Dec

5/4/2023

Programming for Civil Engineers

16

16

Integers in Memory

0 0 1 1 0 1 0 1 0 0 1 1 0 1 0 1

- 2 Byte integer (short)
 - Unsigned short 0 to 65535 ($2^{16}=65536$)
 - Signed short -32768 to 32767 ($2^{15}=32768$)

Integers in Memory

- Unsigned Short
- Unsigned Integer
- Unsigned Long
- Short
- Integer
- Long

Decimals in Memory

- Floating Point

123456789.123456789

19

Scientific Notation

- The Earth's mass is about:

597240000000000000000000000000.0 kg

- In scientific notation:

$$5.9724 \times 10^{24} \text{ kg}$$

- In E (Exp) notation:

5.9724**E**24 kg

- An electron's mass is about

0.00000000000000000000000000000000910938356 kg

9.10938356**E**-31 kg

20

Memory Limitations

- Single variable for large and small numbers with high precision:

$$5.9724E+24 + 9.10938356E-31$$

$$L = 5.9724E+24$$

$$S = 9.10938356E-31$$

$$R1 = L + S$$

$$R2 = R1 - L$$

$$R2 = L + S - L = S ?$$

$$R2 = 0$$

- Select appropriate coordinate system

5/4/2023

Programming for Civil Engineers

21

21

Decimals in Memory

Double precision	010000000000100100100001111101101010100010001000010110100011000
Single precision	0100000001001001000011111011011

64bit = double, double precision



32bit = float, single precision



Signed bit

Exponent

Significand

5/4/2023

Programming for Civil Engineers

22

22

Variable Types

Type	Size	Min	Max
Boolean	Depends!	False	True
Byte	1 byte	0	$255 = 2^8 - 1$
Char	2 bytes	0	$65,535 = 2^{16} - 1$
UShort	2 bytes	0	65,535
UInteger	4 bytes	0	4,294,967,295
ULong	8 bytes	0	18,446,744,073,709,551,615
Integer	4 bytes	-2,147,483,648	2,147,483,647
Long	8 bytes	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
Single	4 bytes	-3.4028235E+38 1.401298E-45	-1.401298E-45 3.4028235E+38
Double	8 bytes	-1.79769313486231570E+308 4.94065645841246544E-324	4.94065645841246544E-324 1.79769313486231570E+308

5/4/2023

Programming for Civil Engineers

23

23

Variable Types

- [Data Type Summary - Visual Basic | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/visual-basic/language-reference/data-types/):

<https://learn.microsoft.com/en-us/dotnet/visual-basic/language-reference/data-types/>

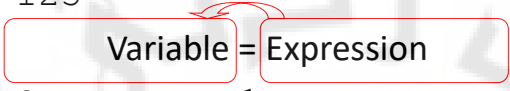
5/4/2023

Programming for Civil Engineers

24

24

Operators

- Assignments operator =
- Counter = 123
- 
- Counter = Counter + 1
- Counter + 1 = 2
- Assignments vs. Equal
 - Counter = Counter + 1
 - 0 = 0 + 1
- Arithmetic Operators: + - × / ^

5/4/2023

Programming for Civil Engineers

25

25

Precedence of Operators

- | | |
|---------------------------|-----|
| 1. Brackets | () |
| 2. Exponentiation | ^ |
| 3. Unary Plus/Minus | + - |
| 4. Multiple / Divide | * / |
| 5. Addition / Subtraction | + - |

5/4/2023

Programming for Civil Engineers

26

26

Declare / Assign Variables

```
Dim intX, intY, intZ As Integer
Dim doubleX, doubleY, doubleZ As Double
doubleX = 1.234
doubleY = doubleX + 5.6
Dim str As String = "Hello"
str = str & " World!"
```

Initialize by default to 0

Double quotation
& concatenate strings

5/4/2023

Programming for Civil Engineers

27

27

Subroutine

- Collect duplicated code in single subroutine
 - DRY: Don't Repeat Yourself
 - Never WET: Write Everything Twice

5/4/2023

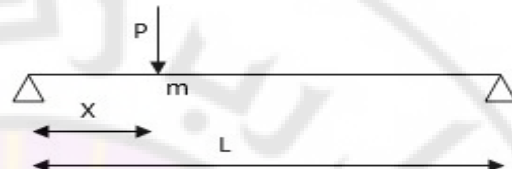
Programming for Civil Engineers

28

28

Example

- Calculate reactions and bending moment



5/4/2023

Programming for Civil Engineers

29

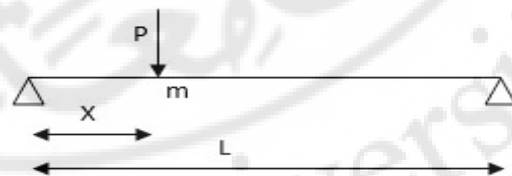
29

Subroutine Syntax

```
Module Program
  Sub SimpleBeamCalculate()
    Dim L, x, P As Double
    Dim RL, RR, m As Double
    L = 3.5 ' m
    x = 1.2 ' m
    P = 5 ' kN

    RR = P * x / L
    RL = P - RR
    m = RR * (L - x)

  End Sub
End Module
```



5/4/2023

Programming for Civil Engineers

30

30

Subroutine Homework

```
Module Program
  Sub SwapVariables()
    Dim Length, Width, Temp As Double
    Length = 4
    Width = 8
    ' Swap values
    Temp = Length ' 4
    Length = Width ' 8
    Width = Temp ' 4
    ' Length = 8
    ' Width = 4
  End Sub
End Module
```

5/4/2023

Programming for Civil Engineers

31

31

Study Material

- Contact:
w@el.sy
- Download material from:
 - <https://pro.el.sy/>
 - <http://damascusuniversity.edu.sy/civil/?lang=1&set=3&id=366>



5/4/2023

Programming for Civil Engineers

32

32

جامعة دمشق
كلية الهندسة المدنية

مادة البرمجة
السنة الثالثة
المحاضرة 3

12/4/2023

Wael Darwich

1

Subroutine Homework

```
Module Program
  Sub SwapVariables()
    Dim Length, Width, Temp As Double
    Length = 4
    Width = 8
    ' Swap values
    Temp = Length ' 4
    Length = Width ' 8
    Width = Temp ' 4
    ' Length = 8
    ' Width = 4
  End Sub
End Module
```

12/4/2023

Programming for Civil Engineers

2

2

Subroutine Homework

```
Module Program
  Sub SwapVariables()
    Dim Length, Width, Temp As Double
    Length = 4
    Width = 8
    ' Swap values
    Length = Length * Width
    Width = Length / Width
    Length = Length / Width
    ' Length = 8
    ' Width = 4
  End Sub
End Module
```

Length	Width	Length	Width
4	8	4	0
32	8	0	0
32	4	0	NaN
8	4	NaN	NaN

12/4/2023

Programming for Civil Engineers

3

3

Subroutine Homework

```
Module Program
  Sub SwapVariables()
    Dim Length, Width As Double
    Length = 4
    Width = 8
    ' Swap values
    Length = Length + Width
    Width = Length - Width
    Length = Length - Width
    ' Length = 8
    ' Width = 4
  End Sub
End Module
```

Length	Width	Length	Width
4	8	4	1.0E+18
12	8	1.0E+18	1.0E+18
12	4	1.0E+18	0
8	4	1.0E+18	0

12/4/2023

Programming for Civil Engineers

4

4

Subroutine Homework

```
Module Program
    Sub SwapVariables()
        Dim Length, Width, Temp As Double
        Length = 4
        Width = 8
        ' Swap values
        Temp = Length ' 4
        Length = Width ' 8
        Width = Temp ' 4
        ' Length = 8
        ' Width = 4
    End Sub
End Module
```

12/4/2023

Programming for Civil Engineers

5

5

Content

- Input & Output (IO)
 - Format
 - Namespace and Imports
- Subroutine & Function
 - Examples
- Conditions
 - Syntax
 - Factorial
 - Recursion

12/4/2023

Programming for Civil Engineers

6

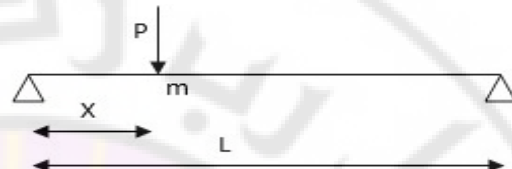
6

Subroutine

```
Module Program
    Sub SimpleBeamCalculate()
        Dim L, x, P As Double
        Dim RL, RR, m As Double
        L = 3.5 ' m
        x = 1.2 ' m
        P = 5 ' kN

        RR = P * x / L
        RL = P - RR
        m = RR * (L - x)

    End Sub
End Module
```



12/4/2023

Programming for Civil Engineers

7

7

IO

```
Module Program
    Sub SimpleBeamCalculate()
        Dim L, x, P As Double
        Dim RL, RR, m As Double
        L = System.Console.ReadLine()
        x = System.Console.ReadLine()
        P = System.Console.ReadLine()

        RR = P * x / L
        RL = P - RR
        m = RR * (L - x)
        System.Console.WriteLine(m)

    End Sub
End Module
```

12/4/2023

Programming for Civil Engineers

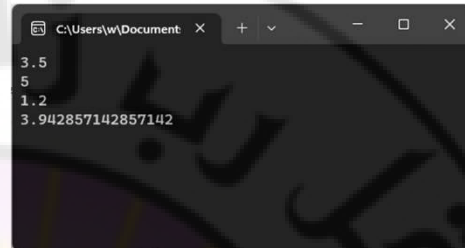
8

8

IO

```
Imports System.Console
Module Program
    Sub SimpleBeamCalculate()
        Dim L, x, P As Double
        Dim RL, RR, m As Double
        L = ReadLine()
        x = ReadLine()
        P = ReadLine()

        RR = P * x / L
        RL = P - RR
        m = RR * (L - x)
        WriteLine(m)
    End Sub
End Module
```



```
C:\Users\W\Document x + - □ x
3.5
5
1.2
3.942857142857142
```

12/4/2023

Programming for Civil Engineers

9

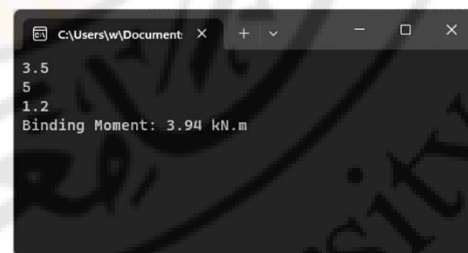
9

IO

```
WriteLine("Binding Moment: ")
WriteLine(m)

WriteLine("Binding Moment: " & m & " kN.m")

WriteLine("Binding Moment: " & System.Math.Round(m, 2) & " kN.m")
```



```
C:\Users\W\Document x + - □ x
3.5
5
1.2
Binding Moment: 3.94 kN.m
```

12/4/2023

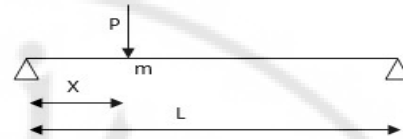
Programming for Civil Engineers

10

10

10

```
Imports System.Console
Imports System.Math
Module Program
Sub Main()
    Dim L, x, P, RR, RL, m As Double
    WriteLine("Enter Beam Length [m] ")
    L = ReadLine()
    WriteLine("Enter Load Value [kN] ")
    P = ReadLine()
    WriteLine("Enter Load distance from left [m] ")
    x = ReadLine()
    RR = P * x / L
    RL = P - RR
    m = RR * (L - x)
    WriteLine("Right Reaction= " & Round(RR, 2) & " kN")
    WriteLine("Left Reaction= " & Round(RL, 2) & " kN")
    WriteLine("Bending Moment= " & Round(m, 2) & " kN.m")
End Sub
End Module
```



```
C:\Users\w\Document x + - □ x
Enter Beam Length [m]
3.5
Enter Load Value [kN]
5
Enter Load distance from left [m]
1.2
Right Reaction= 1.71 kN
Left Reaction= 3.29 kN
Bending Moment= 3.94 kN.m
```

12/4/2023

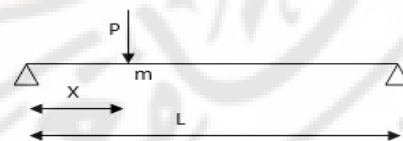
Programming for Civil Engineers

11

11

10

```
Imports System.Console
Imports System.Math
Module Program
Sub Main()
    Dim L, x, P, RR, RL, m As Double
    WriteLine("Enter Beam Length [m] ")
    L = ReadLine()
    WriteLine("Enter Load Value [kN] ")
    P = ReadLine()
    WriteLine("Enter Load distance from left [m] ")
    x = ReadLine()
    RR = P * x / L
    RL = P - RR
    m = RR * (L - x)
    WriteLine("RL RR")
    WriteLine(Round(RL, 2) & " " & Round(RR, 2))
End Sub
End Module
```



```
C:\Users\w\Document x + - □ x
Enter Beam Length [m]
3.5
Enter Load Value [kN]
5
Enter Load distance from left [m]
1.2
RL RR
3.29 1.71
```

12/4/2023

Programming for Civil Engineers

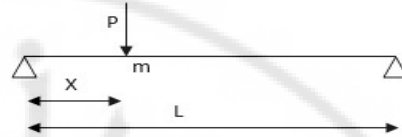
12

12

IO

```
Imports System.Console
Imports System.Math
Module Program
Sub Main()
    Dim L, x, P, RR, RL, m As Double
    WriteLine("Enter Beam Length [m] ")
    L = ReadLine()
    WriteLine("Enter Load Value [kN] ")
    P = ReadLine()
    WriteLine("Enter Load distance from left [m] ")
    x = ReadLine()
    RR = P * x / L
    RL = P - RR
    m = RR * (L - x)
    WriteLine("RL" & Chr(9) & "RR")
    WriteLine(Round(RL, 2) & Chr(9) & Round(RL, 2))

End Sub
End Module
```



A screenshot of a console window titled "C:\Users\w\Document" showing the program's execution. The user has entered 3.5 for beam length, 5 for load value, and 1.2 for load distance. The output shows RL = 3.29 and RR = 1.71.

```
Enter Beam Length [m]
3.5
Enter Load Value [kN]
5
Enter Load distance from left [m]
1.2
RL      RR
3.29    1.71
```

12/4/2023

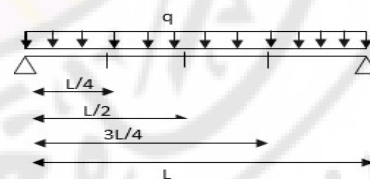
Programming for Civil Engineers

13

13

Table Format Example

- Inputs:
 - q
 - L
- Calculate bending moment:
 - At 3 points
 - $0.25 L$
 - $0.50 L$
 - $0.75 L$



12/4/2023

Programming for Civil Engineers

14

14

```
Imports System.Console
Imports System.Math
Module Program
```

```
Sub Main()
```

```
Dim q, L, M1, M2, M3 As Double
```

```
WriteLine("Enter beam length in [m] ")
```

```
L = ReadLine()
```

```
WriteLine("Enter load value in [kN/m] ")
```

```
q = ReadLine()
```

```
M2 = q * L ^ 2 / 8
```

```
M1 = M2 * 3 / 4
```

```
M3 = M1
```

```
WriteLine("Bending Moments")
```

```
WriteLine("M1" & Chr(9) & "M2" & Chr(9) & "M3")
```

```
WriteLine("---" & Chr(9) & "---" & Chr(9) & "---")
```

```
WriteLine(Round(M1, 2) & Chr(9) & Round(M2, 2) & Chr(9) & Round(M3, 2))
```

```
End Sub
```

```
End Module
```

```
Microsoft Visual Studi x + - □ x
Enter beam length in [m]
4.5
Enter load value in [kN/m]
3
Bending Moments
M1      M2      M3
--      --      --
5.7      7.59    5.7
```

12/4/2023

Programming for Civil Engineers

15

15

Namespace.Module.Function

- Same function names:
 - Avoid name conflict
- Module
- Namespace
- Examples
 - System.Console.WriteLine
 - System.Math.Round
- Import System.Math
 - Round

12/4/2023

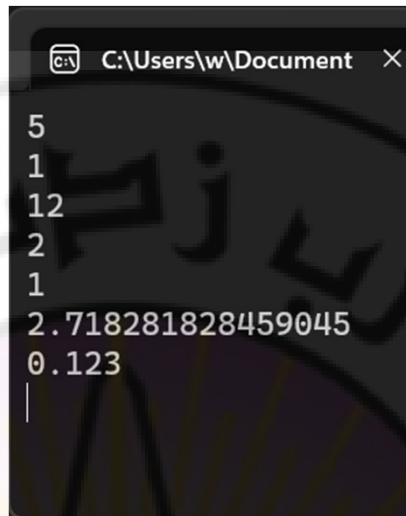
Programming for Civil Engineers

16

16

System.Math

```
WriteLine(Sqrt(25))
WriteLine(Sin(PI / 2))
WriteLine(Abs(-12))
WriteLine(Log10(100))
WriteLine(Log(E))
WriteLine(Exp(1))
WriteLine(Round(0.12345, 3))
```



12/4/2023

Programming for Civil Engineers

17

17

Distance Subroutine

```
Imports System.Console
Imports System.Math
Module Program
    Sub Distance()
        Dim x1, y1, x2, y2, dist As Double
        WriteLine("Enter X1")
        x1 = ReadLine()
        WriteLine("Enter Y1")
        y1 = ReadLine()
        WriteLine("Enter X2")
        x2 = ReadLine()
        WriteLine("Enter Y2")
        y2 = ReadLine()
        dist = Sqrt((x2 - x1) ^ 2 + (y2 - y1) ^ 2)
        WriteLine("Distance = " & Round(dist, 2))
    End Sub
End Module
```

12/4/2023

Programming for Civil Engineers

18

18

Distance Subroutine

```
Module Program
    Sub Distance(x1 As Double, y1 As Double, x2 As Double, y2 As Double)
        Dim x1, y1, x2, y2, dist As Double

        x1 = ReadLine()
        y1 = ReadLine()
        x2 = ReadLine()
        y2 = ReadLine()

        dist = Sqrt((x2 - x1) ^ 2 + (y2 - y1) ^ 2)
        WriteLine("Distance = " & Round(dist, 2))
    End Sub
End Module
```

12/4/2023

Programming for Civil Engineers

19

19

Distance Subroutine

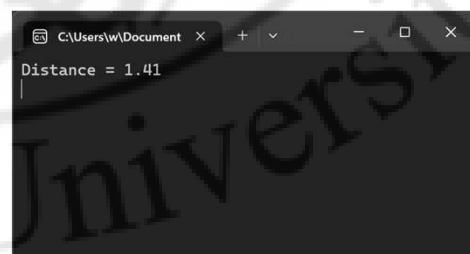
```
Module Program
    Sub Distance(x1 As Double, y1 As Double, x2 As Double, y2 As Double)
        Dim dist As Double
        dist = Sqrt((x2 - x1) ^ 2 + (y2 - y1) ^ 2)
        WriteLine("Distance = " & Round(dist, 2))
    End Sub

    Sub Main()
        Distance(0, 0, 1, 1)
    End Sub
End Module
```

12/4/2023

Programming for Civil Engineers

20



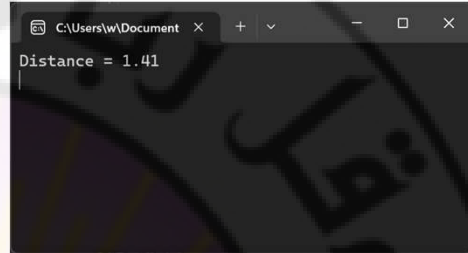
20

Distance Function

```
Module Program
    Function Distance(x1 As Double, y1 As Double, x2 As Double, y2 As Double) As Double
        Dim dist As Double
        dist = Sqrt((x2 - x1) ^ 2 + (y2 - y1) ^ 2)
        Return dist
    End Function

    Sub Main()
        Dim d As Double = Distance(0, 0, 1, 1)
        WriteLine("Distance = " & Round(d, 2))
    End Sub

End Module
```



12/4/2023

Programming for Civil Engineers

21

21

Function Syntax

```
Function fx(var1 As Double, var2 As Double) As Double
    Dim result As Double
    ' Calcs
    Return result
End Function
```

12/4/2023

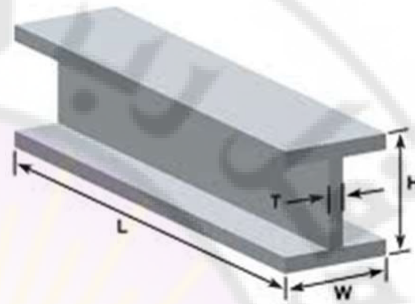
Programming for Civil Engineers

22

22

Examples

- For the beam shown below, write VB subroutine to calculate its cross-section area and volume



12/4/2023

Programming for Civil Engineers

23

23

```
Imports System.Console
Module IBeam
    Sub Main()
        Dim W, T, H, L, SA, Vol As Double
        WriteLine("Enter Beam Section Width in [mm] ")
        W = ReadLine()
        WriteLine("Enter Beam Section Height in [mm] ")
        H = ReadLine()
        WriteLine("Enter Beam Section Thickness in [mm] ")
        T = ReadLine()
        WriteLine("Enter Beam Length in m: ")
        L = ReadLine()
        SA = (2 * W * T + (H - 2 * T) * T) / 100
        Vol = (SA / 10000) * L
        WriteLine("Section Area = " & SA & " cm2")
        WriteLine("Volume = " & Vol & " m3")
    End Sub
End Module
```

12/4/2023

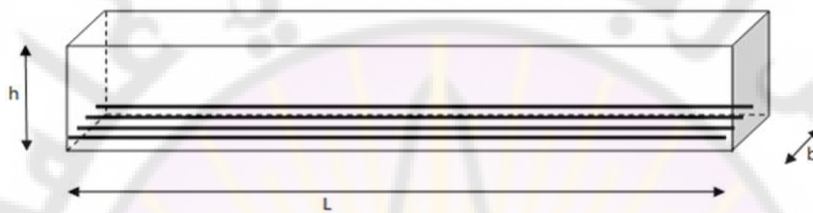
Programming for Civil Engineers

24

24

Examples

- For the beam shown below, write VB subroutine to calculate:
 - Concrete volume
 - Reinforcement weight, for n bar, at length L+20cm



12/4/2023

Programming for Civil Engineers

25

25

```
Imports System.Console
Imports System.Math
Module Calc
    Sub Main()
        Dim L, b, h, d As Double
        WriteLine("Enter Beam Length [m] ")
        L = ReadLine()
        WriteLine("Enter Section Width [cm] ")
        b = ReadLine()
        WriteLine("Enter Section Height [cm] ")
        h = ReadLine()
        WriteLine("Enter bars diameter [mm] ")
        d = ReadLine()

        Dim n As Integer
        WriteLine("Enter number of bars ")
        n = ReadLine()

        Dim V As Double = L * b / 100 * h / 100
        Dim steel As Double = 7850 ' Steel density kg/m3
        Dim W As Double = PI * (d / 1000) ^ 2 / 4 * (L + 0.2) * n * steel

        WriteLine("Concrete Volume= " & V & " m3")
        WriteLine("Reinforcement Weight= " & W & " kg")
    End Sub
End Module
```

12/4/2023

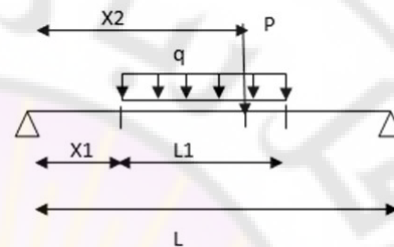
Programming for Civil Engineers

26

26

Examples

- For the beam shown below, write VB subroutine to calculate support reactions



12/4/2023

Programming for Civil Engineers

27

27

```
Imports System.Console
Module Module1
    Sub Main()
        Dim q, P, L, X1, X2, L1, RR, RL As Double
        WriteLine("Enter beam length in [m] ")
        L = ReadLine()
        WriteLine("Enter distributed load value in [t/m] ")
        q = ReadLine()
        WriteLine("Enter q distance from left [m] ")
        X1 = ReadLine()
        WriteLine("Enter q length [m] ")
        L1 = ReadLine()
        WriteLine("Enter P load value in [t] ")
        P = ReadLine()
        WriteLine("Enter P distance from left [m] ")
        X2 = ReadLine()

        RR = (P * X2 + q * L1 * (X1 + L1 / 2)) / L
        RL = P + q * L1 - RR

        WriteLine("Right Reaction = " & RR & " t")
        WriteLine("Left Reaction = " & RL & " t")
    End Sub
End Module
```

12/4/2023

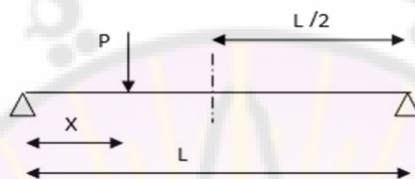
Programming for Civil Engineers

28

28

Examples

- For the beam shown below, write VB subroutine to calculate shear force and bending moment in the middle



12/4/2023

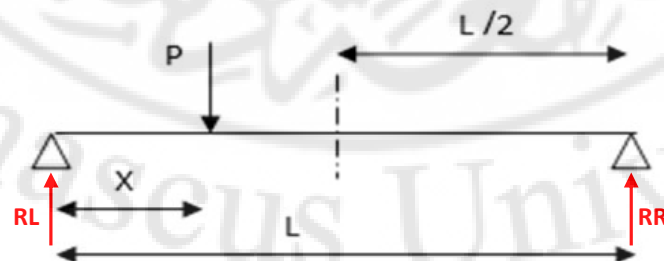
Programming for Civil Engineers

29

29

Conditions

$$M = RR * L / 2$$



12/4/2023

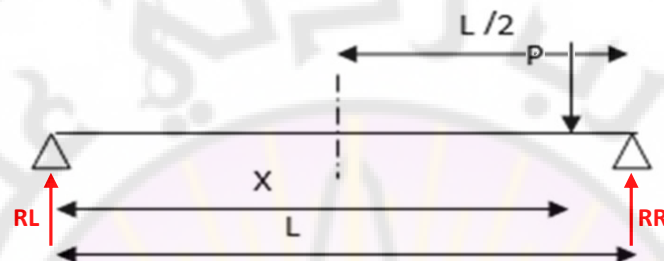
Programming for Civil Engineers

30

30

Conditions

$$M = RR * L / 2 - P * (X - L / 2)$$



12/4/2023

Programming for Civil Engineers

31

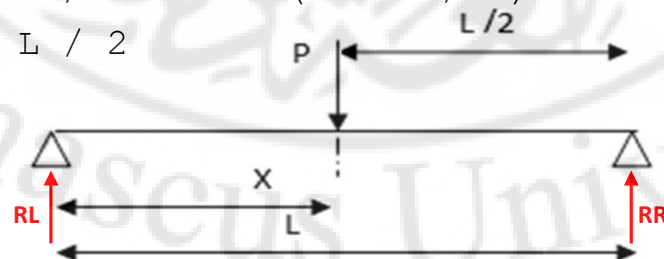
31

Conditions

$$X = L / 2$$

$$M = RR * L / 2 - P * (X - L / 2)$$

$$M = RR * L / 2$$



12/4/2023

Programming for Civil Engineers

32

32

```

Dim P, L, X, RR, M, Q As Double
WriteLine("Enter the length of the beam [m] ")
L = ReadLine()
WriteLine("Enter the load value [N] ")
P = ReadLine()
WriteLine("Enter the load distance from the left [m] ")
X = ReadLine()
RR = P * X / L

If (X < L / 2) Then
    M = RR * L / 2
    Q = RR
Else
    M = RR * L / 2 - P * (X - L / 2)
    Q = RR - P
End If

WriteLine("Bending Moment = " & Round(M, 2) & " N.m")
WriteLine("Shear Force = " & Round(Q, 2) & " N")

```

12/4/2023

Programming for Civil Engineers

33

33

If Syntax

```

If condition Then
    ' Do A
Else
    ' Do B
End If

```

12/4/2023

Programming for Civil Engineers

34

34

Relational Operators

- Numerical comparison:
 - $A < B$ Smaller
 - $A \leq B$ Smaller equal
 - $A > B$ Greater
 - $A \geq B$ Greater equal
 - $A = B$ Equal
 - $A \neq B$ Not equal
- Not
- And
- Or

12/4/2023

Programming for Civil Engineers

35

35

Relational Operators

Not: $\text{False} = \text{Not True}$

And: $b > a$ And $b < c$

$\text{True And True} = \text{True}$

$\text{False And False} = \text{False}$

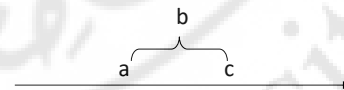
$\text{True And False} = \text{False}$ And $\text{True} = \text{False}$

Or: $b < a$ Or $b > c$

$\text{True Or True} = \text{True}$

$\text{False Or False} = \text{False}$

$\text{True Or False} = \text{False}$ Or $\text{True} = \text{True}$



12/4/2023

Programming for Civil Engineers

36

36

Relational Operators

```
Dim a, b, c, d As Integer
a = 1
b = 2
c = 3
d = 4
Dim test As Boolean
test = a < b And b < c
      True And True
test = a < b Or d < b And d < c
      1 + 2 × 3
      True Or False And False
      True Or (False And False) = True
      (True Or False) And False = False
```

12/4/2023

Programming for Civil Engineers

37

37

Operator Precedence

1. Brackets
2. Not
3. And
4. Or

12/4/2023

Programming for Civil Engineers

38

38

Factorial Example

$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$$

$$n! = n \times (n-1)!$$

$$n > 1$$

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

$$6! = 6 \times 5!$$

$$2! = 2 \times 1!$$

$$1! = 1$$

12/4/2023

Programming for Civil Engineers

39

39

Factorial Function

```
Function Factorial(n As Integer) As Integer
    If n > 1 Then
        Return n * Factorial(n - 1)
    Else
        Return 1
    End If
End Function
```

12/4/2023

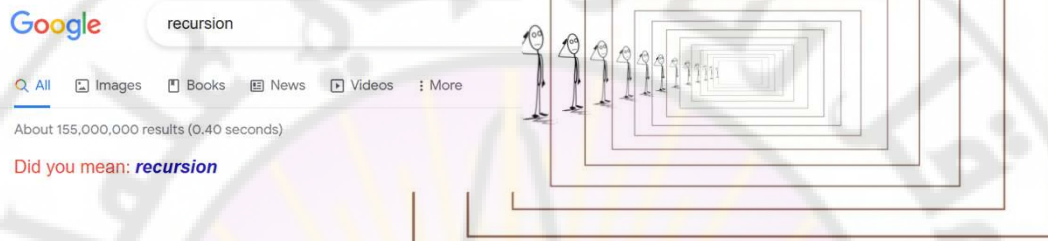
Programming for Civil Engineers

40

40

Recursive Functions

- When a function call itself



جامعة دمشق
كلية الهندسة المدنية

مادة البرمجة
السنة الثالثة
المحاضرة 4

3/5/2023

Wael Darwich

1

Content

- Condition Syntax
 - If / End If
 - Then
 - Else
 - Elself
- Examples
 - Section Properties
 - Equation Solver
 - Previous Exam
 - Input Validation
 - Electricity Tariff
- Loops

3/5/2023

Programming for Civil Engineers

2

2

If Syntax

```
If condition1 Then
    code1
End If
```

```
If condition2 Then
    code2
End If
```

3/5/2023

Programming for Civil Engineers

3

3

If Syntax

```
If condition1 Then
    code1
    If condition2 Then
        code2
    End If
End If
```

3/5/2023

Programming for Civil Engineers

4

4

Single Line

```
If condition1 Then code1
```

No need for `End If`

Any command after `Then`

If Syntax

```
If condition1 Then  
    code1 ' True  
Else  
    code2 ' False  
End If
```

If Syntax

```

If condition1 Then
    code1 ' Con1
ElseIf condition2 Then
    code2 ' Con2 and not Con1
ElseIf condition3 Then
    code3 ' Con3 and not Con1 and not Con2
Else
    ' Last one
    code4 ' not Con1 and not Con2 and not Con3
End If

```

3/5/2023

Programming for Civil Engineers

7

7

If Syntax

```

{ If condition1 Then
    code1
  End If

{ If condition2 Then
    code2
  End If

```

3/5/2023

Programming for Civil Engineers

8

8

If Syntax

1. `If` condition1 `Then`
2. `ElseIf` any number ≤ 0
3. `Else` optional
4. `End If`
5. Exact order

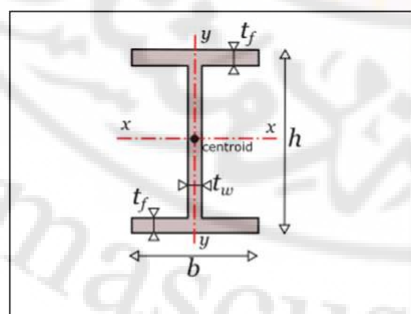
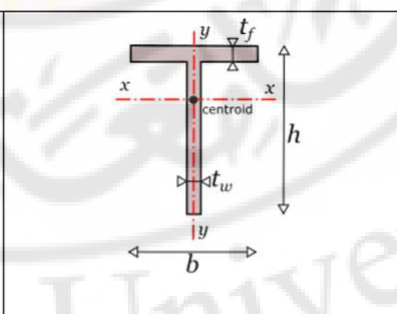
3/5/2023

Programming for Civil Engineers

9

9

Example: Section Properties

	
$A = 2 b \cdot t_f + (h - 2 \cdot t_f) \cdot t_w$ $I_y = \frac{(h - 2t_f)t_w^3}{12} + 2 \frac{t_f b^3}{12}$	$A = b \cdot t_f + (h - t_f) \cdot t_w$ $I_y = \frac{(h - t_f)t_w^3}{12} + \frac{t_f b^3}{12}$

3/5/2023

Programming for Civil Engineers

10

10

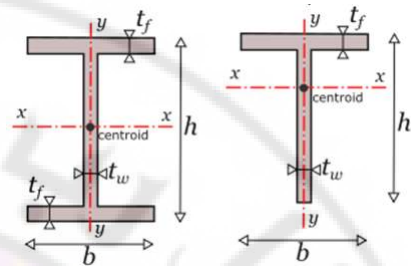
Example: Section Properties

```
Imports System.Console
Imports System.Math

Module section
    Sub main()

        Dim b, h, tf, tw, A, type As Integer
        Dim Iy As Double

    End Sub
End Module
```



3/5/2023

Programming for Civil Engineers

11

11

Example: Section Properties

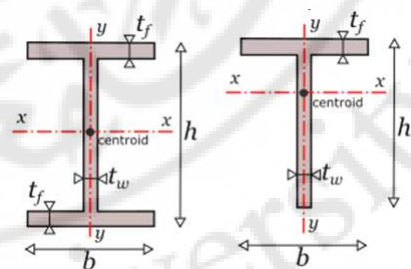
```
WriteLine("Enter section type (I=1/T=2)")
type = ReadLine()

WriteLine("Enter tf (mm)")
tf = ReadLine()

WriteLine("Enter tw (mm)")
tw = ReadLine()

WriteLine("Enter b (mm)")
b = ReadLine()

WriteLine("Enter h (mm)")
h = ReadLine()
```



3/5/2023

Programming for Civil Engineers

12

12

Example: Section Properties

```

If (type = 1) Then
    A = 2 * tf * b + (h - 2 * tf) * tw
    Iy = 2 * tf * b ^ 3 / 12 + (h - 2 * tf) * tw ^ 3 / 12
ElseIf (type = 2) Then
    A = tf * b + (h - tf) * tw
    Iy = tf * b ^ 3 / 12 + (h - tf) * tw ^ 3 / 12
End If

WriteLine("Area = " & A & " mm2")
WriteLine("Iy = " & Round(Iy, 2) & " mm4")

```

$$A = 2 \cdot b \cdot t_f + (h - 2 \cdot t_f) \cdot t_w$$

$$I_y = \frac{(h - 2t_f)t_w^3}{12} + 2 \frac{t_f b^3}{12}$$

$$A = b \cdot t_f + (h - t_f) \cdot t_w$$

$$I_y = \frac{(h - t_f)t_w^3}{12} + \frac{t_f b^3}{12}$$

3/5/2023

Programming for Civil Engineers

13

13

Example: Equation Solver

- Quadratic (second order) equations:

$$A \cdot x^2 + B \cdot x + C = 0$$

- Inputs:

- A, B, C

- Outputs:

- X_1, X_2

3/5/2023

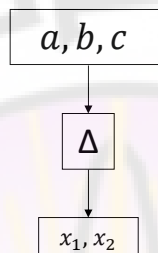
Programming for Civil Engineers

14

14

Quadratic Equations

$$a \cdot x^2 + b \cdot x + c = 0$$



3/5/2023

Programming for Civil Engineers

15

15

Quadratic Equations

$$a \cdot x^2 + b \cdot x + c = 0$$

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2 \cdot a}$$

$$a = 0$$

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{0}$$

3/5/2023

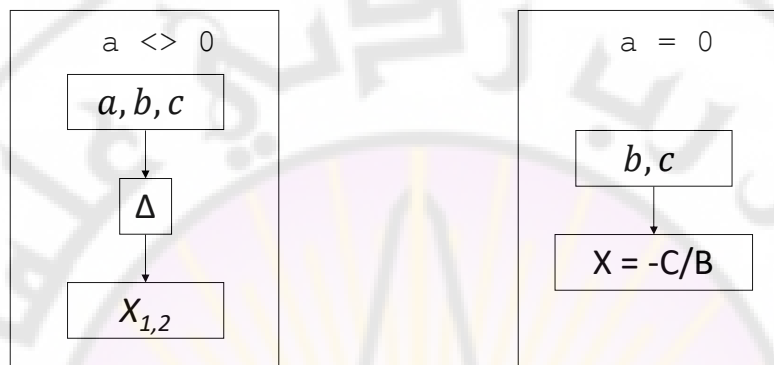
Programming for Civil Engineers

16

16

Quadratic Equations

$$a \cdot x^2 + b \cdot x + c = 0$$



3/5/2023

Programming for Civil Engineers

17

17

Error Check

$$a \cdot x^2 + b \cdot x + c = 0$$

$$a = 0$$

$$b = 0$$

No solution!

Input validation

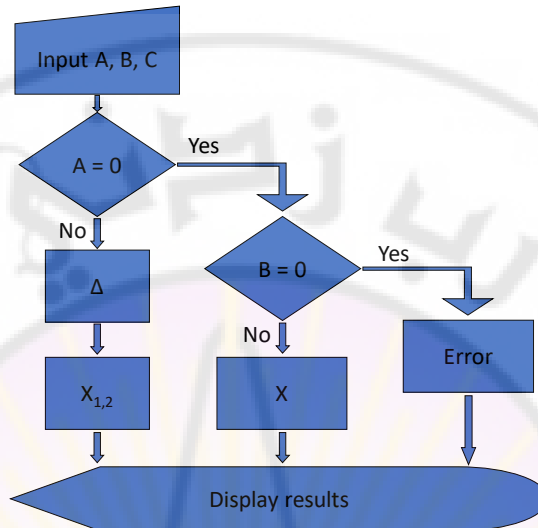
3/5/2023

Programming for Civil Engineers

18

18

Flowchart



3/5/2023

Programming for Civil Engineers

19

19

```

Imports System.Console
Imports System.Math

Module Program

    Sub QuadraticEquationRoots(a As Double, b As Double, c As Double)
        ' Solve second order equation a x^2 + b x + c = 0

    End Sub

    Sub main()
        ' Solve quadratic equations
        QuadraticEquationRoots(1, 0, -4)
    End Sub

End Module
  
```

20

```

If a = 0 Then
    If b = 0 Then ' Regardless of c
        WriteLine("Error: This equation is degenerate!")
    Else
        WriteLine("There is a single root = " & Round(-c / b, 3))
    End If
Else
    Dim delta As Double = b ^ 2 - 4 * a * c
    If delta >= 0 Then
        Dim x1, x2, delta2 As Double
        delta2 = Sqrt(delta)
        x1 = (-b + delta2) / (2 * a)
        x2 = (-b - delta2) / (2 * a)
        WriteLine("There are 2 roots x1=" & Round(x1, 3) & " x2=" & Round(x2, 3))
    Else
        WriteLine("There are 2 imaginary roots for this equation")
    End If
End If

```

21

```

If a = 0 Then
    If b = 0 Then ' Regardless of c
        WriteLine("Error: This equation is degenerate!")
    Else
        WriteLine("There is a single root = " & Round(-c / b, 3))
    End If
Else
    Dim delta As Double = b ^ 2 - 4 * a * c
    If delta >= 0 Then
        Dim x1, x2, delta2 As Double
        delta2 = Sqrt(delta)
        x1 = (-b + delta2) / (2 * a)
        x2 = (-b - delta2) / (2 * a)
        WriteLine("There are 2 roots x1=" & Round(x1, 3) & " x2=" & Round(x2, 3))
    Else
        WriteLine("There are 2 imaginary roots for this equation")
    End If
End If

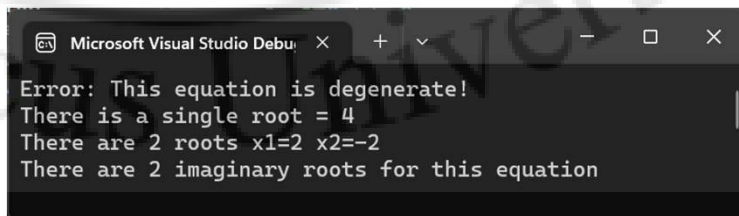
```

22

```
Imports System.Console
Imports System.Math
Module Program
    Sub QuadraticEquationRoots(a As Double, b As Double, c As Double)
        ' Solve second order equation a x^2 + b x + c = 0
        If a = 0 Then
            If b = 0 Then ' Regardless of c
                WriteLine("Error: This equation is degenerate!")
            Else
                WriteLine("There is a single root = " & Round(-c / b, 3))
            End If
        Else
            Dim delta As Double = b ^ 2 - 4 * a * c
            If delta >= 0 Then
                Dim x1, x2, delta2 As Double
                delta2 = Sqrt(delta)
                x1 = (-b + delta2) / (2 * a)
                x2 = (-b - delta2) / (2 * a)
                WriteLine("There are 2 roots x1=" & Round(x1, 3) & " x2=" & Round(x2, 3))
            Else
                WriteLine("There are 2 imaginary roots for this equation")
            End If
        End If
    End Sub
    Sub main()
        ' Solve quadratic equations
        QuadraticEquationRoots(0, 0, 0)
        QuadraticEquationRoots(0, 1, -4)
        QuadraticEquationRoots(1, 0, -4)
        QuadraticEquationRoots(1, 0, 4)
    End Sub
End Module
```

23

```
Sub main()
    ' Solve quadratic equations
    QuadraticEquationRoots(0, 0, 0)
    QuadraticEquationRoots(0, 1, -4)
    QuadraticEquationRoots(1, 0, -4)
    QuadraticEquationRoots(1, 0, 4)
End Sub
```



```
Microsoft Visual Studio Debu  × + - □ ×
Error: This equation is degenerate!
There is a single root = 4
There are 2 roots x1=2 x2=-2
There are 2 imaginary roots for this equation
```

24

Example: Previous Exam

```
Dim X, Y As Double
WriteLine("Enter X then Y:")
X = ReadLine()
Y = ReadLine()
```

اكتب برنامج بلغة VB.Net لتحديد
موقع نقطة $P(x, y)$ ضمن جملة
احداثيات ديكارتية

3/5/2023

Programming for Civil Engineers

25

25

Example: Previous Exam

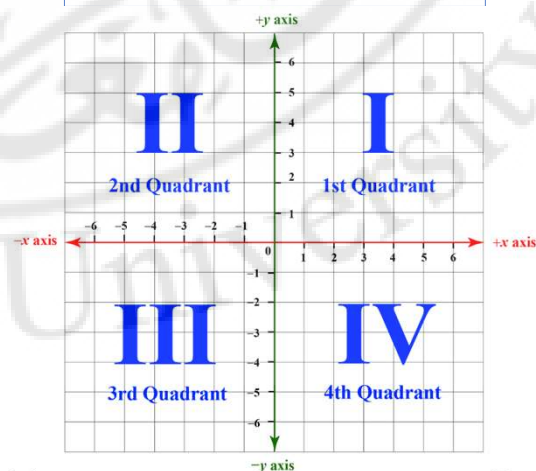
```
If X = 0 And Y = 0 Then
    WriteLine("origin point")
ElseIf X = 0 Then
    WriteLine("on axis Y")
ElseIf Y = 0 Then
    WriteLine("on axis X")
ElseIf X > 0 And Y > 0 Then
    WriteLine("in 1st quarter")
ElseIf X < 0 And Y > 0 Then
    WriteLine("in 2nd quarter")
ElseIf X < 0 And Y < 0 Then
    WriteLine("in 3rd quarter")
Else
    WriteLine("in 4th quarter")
End If
```

3/5/2023

Programming for Civil Engineers

26

اكتب برنامج بلغة VB.Net لتحديد
موقع نقطة $P(x, y)$ ضمن جملة
احداثيات ديكارتية



26

Example: Previous Exam

```

If X = 0 And Y = 0 Then
    WriteLine("origin point")
ElseIf X = 0 Then
    WriteLine("on axis Y")
ElseIf Y = 0 Then
    WriteLine("on axis X")
ElseIf X > 0 And Y > 0 Then
    WriteLine("in 1st quarter")
ElseIf X < 0 And Y > 0 Then
    WriteLine("in 2nd quarter")
ElseIf X < 0 And Y < 0 Then
    WriteLine("in 3rd quarter")
Else
    WriteLine("in 4th quarter")
End If

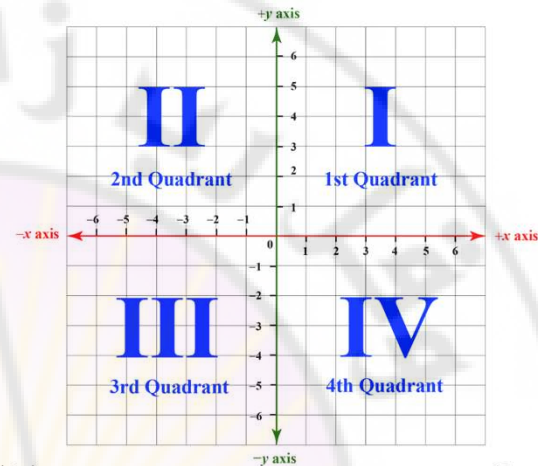
```

3/5/2023

Programming for Civil Engineers

27

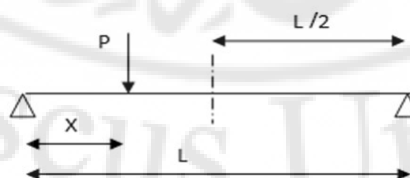
اكتب برنامج بلغة VB.Net لتحديد
موقع نقطة $P(x, y)$ ضمن جملة
احداثيات ديكارتية



27

Input Validation

- For the beam shown below, write VB subroutine to calculate shear force and bending moment in the middle



3/5/2023

Programming for Civil Engineers

28

28

Input Validation

```

Dim P, L, X, RR, M, Q As Double
WriteLine("Enter the length of the beam [m] ")
L = ReadLine()
WriteLine("Enter the load value [N] ")
P = ReadLine()
WriteLine("Enter the load distance from the left [m] ")
X = ReadLine()
RR = P * X / L

If (X < L / 2) Then
    M = RR * L / 2
    Q = RR
Else
    M = RR * L / 2 - P * (X - L / 2)
    Q = RR - P
End If

WriteLine("Bending Moment = " & Round(M, 2) & " N.m")
WriteLine("Shear Force = " & Round(Q, 2) & " N")

```

3/5/2023

Programming for Civil Engineers

29

29

Input Validation

```

Dim P, L, X, RR, M, Q As Double
WriteLine("Enter the length of the beam [m] ")
L = ReadLine()
WriteLine("Enter the load value [N] ")
P = ReadLine()
WriteLine("Enter the load distance from the left [m] ")
X = ReadLine()

If (L <= 0 Or X < 0 Or X > L) Then
    WriteLine("Input Error!")
Else
    RR = P * X / L
    If (X < L / 2) Then

    End If
End If

```



3/5/2023

Programming for Civil Engineers

30

30

Example: Electricity Tariff

التعرفة:

من ١ إلى ٦٠٠ بـ ٢ ل.ب، من ٦٠١ إلى ١٠٠٠ بـ ٦ ل.ب، من ١٠٠١ إلى ١٥٠٠ بـ ٢٠ ل.ب، من ١٥٠١ إلى ٢٥٠٠ بـ ٩٠ ل.ب، من ٢٥٠١ فما فوق بـ ١٥٠ ل.ب لكل ك.وس

```
Function tariff(kwh As Integer) As Integer
    Dim result As Integer

    Return result
End Function
```

3/5/2023

Programming for Civil Engineers

31

31

Example: Electricity Tariff

التعرفة:

من ١ إلى ٦٠٠ بـ ٢ ل.ب، من ٦٠١ إلى ١٠٠٠ بـ ٦ ل.ب، من ١٠٠١ إلى ١٥٠٠ بـ ٢٠ ل.ب، من ١٥٠١ إلى ٢٥٠٠ بـ ٩٠ ل.ب، من ٢٥٠١ فما فوق بـ ١٥٠ ل.ب لكل ك.وس

```
If kwh < 1 Then
    result = 0
ElseIf kwh <= 600 Then
    result = kwh * 2
ElseIf kwh <= 1000 Then
    result = 600 * 2 + (kwh - 600) * 6
ElseIf kwh <= 1500 Then
    result = 600 * 2 + 400 * 6 + (kwh - 1000) * 20
ElseIf kwh <= 2500 Then
    result = 600 * 2 + 400 * 6 + 500 * 20 + (kwh - 1500) * 90
Else
    result = 600 * 2 + 400 * 6 + 500 * 20 + 1000 * 90 + (kwh - 2500) * 150
End If
```

3/5/2023

Programming for Civil Engineers

32

32

Example: Electricity Tariff

التعرفة:

من ١ إلى ٦٠٠ بـ ٢ ل.ب، من ٦٠١ إلى ١٠٠٠ بـ ٦ ل.ب، من ١٠٠١ إلى ١٥٠٠ بـ ٢٠ ل.ب، من ١٥٠١ إلى ٢٥٠٠ بـ ٩٠ ل.ب، من ٢٥٠١ فما فوق بـ ١٥٠ ل.ب لكل ك.وس

```
Sub main()
    WriteLine("Cost of 100 kwh is " & tariff(100))
    WriteLine("Cost of 1000 kwh is " & tariff(1000))
    WriteLine("Cost of 1100 kwh is " & tariff(1100))
End Sub
```



```
Microsoft Visual Studio
Cost of 100 kwh is 200
Cost of 1000 kwh is 3600
Cost of 1100 kwh is 5600
```

3/5/2023

Programming for Civil Engineers

33

33

Example: Electricity Tariff

التعرفة:

من ١ إلى ٦٠٠ بـ ٢ ل.ب، من ٦٠١ إلى ١٠٠٠ بـ ٦ ل.ب، من ١٠٠١ إلى ١٥٠٠ بـ ٢٠ ل.ب، من ١٥٠١ إلى ٢٥٠٠ بـ ٩٠ ل.ب، من ٢٥٠١ فما فوق بـ ١٥٠ ل.ب لكل ك.وس

If kwh < 1 Then	If kwh > 1 Then	If kwh > 2500 Then
ElseIf kwh <= 600 Then	ElseIf kwh > 600 Then	ElseIf kwh > 1500 Then
ElseIf kwh <= 1000 Then	ElseIf kwh > 1000 Then	ElseIf kwh > 1000 Then
ElseIf kwh <= 1500 Then	ElseIf kwh > 1500 Then	ElseIf kwh > 600 Then
ElseIf kwh <= 2500 Then	ElseIf kwh > 2500 Then	ElseIf kwh > 1 Then
Else	Else	Else
End If	End If	End If

3/5/2023

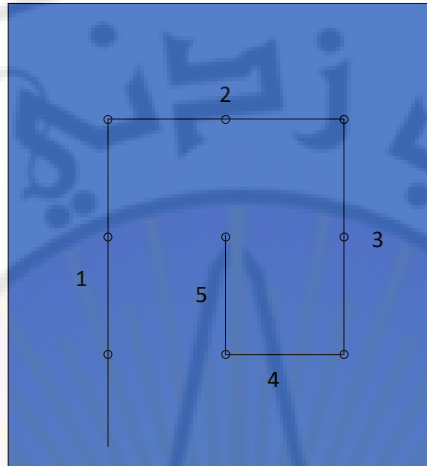
Programming for Civil Engineers

34

34

Line Puzzle

Connect all nine dots with five straight lines without lifting up your pen.
Connect all nine dots with FOUR straight lines without lifting up your pen



3/5/2023

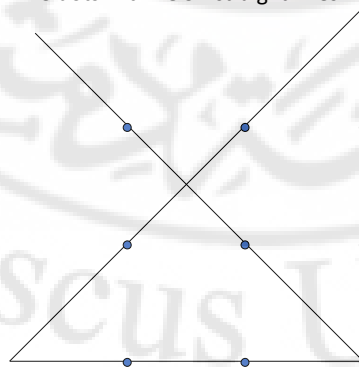
Programming for Civil Engineers

35

35

Line Puzzle

Connect all nine dots with FOUR straight lines without lifting your pencil



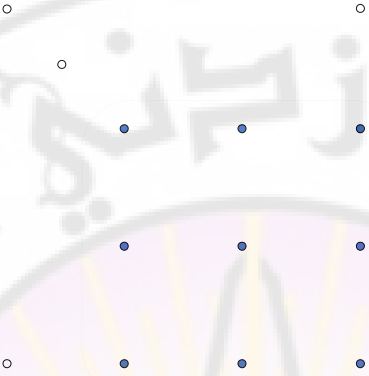
3/5/2023

Programming for Civil Engineers

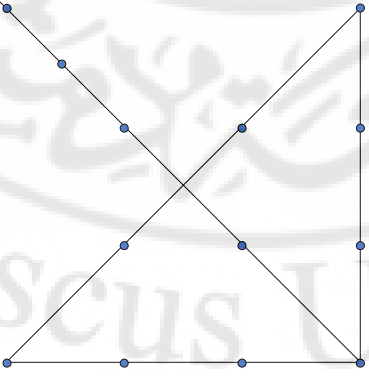
36

36

Line Puzzle



Line Puzzle



Bottle Problem

- Half full / half empty?
- Calculate volume
- Iterations
 - Loop



3/5/2023

39

جامعة دمشق
كلية الهندسة المدنية

مادة البرمجة
السنة الثالثة
المحاضرة 5

10/5/2023

Wael Darwich

1

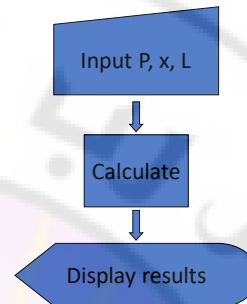
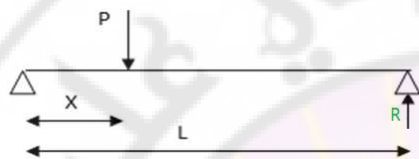
Content

- Loops Types
- Syntax
- Examples
- Last Lecture
 - Programming Project
 - w@el.sy

2

Program Commands 1

- Calculate reaction
 - Single path / sequence of commands



10/5/2023

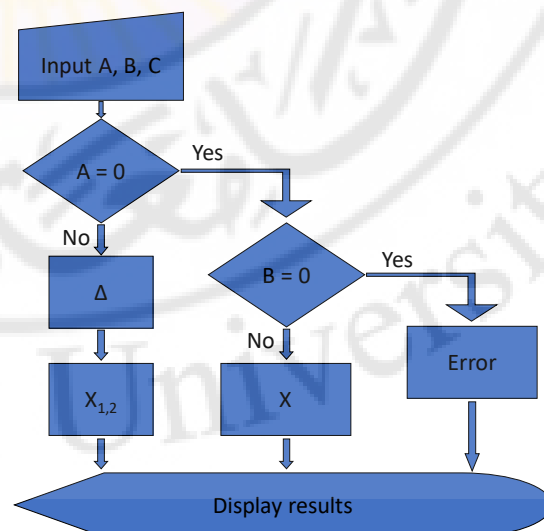
Programming for Civil Engineers

3

3

Program Commands 2

- Second order equation
 - Check for conditions
 - Select a path



10/5/2023

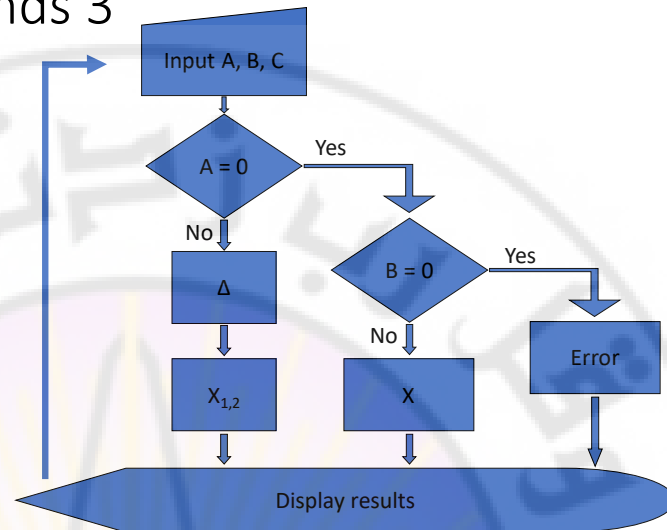
Programming for Civil Engineers

4

4

Program Commands 3

- Repeat
 - Start again / loop



10/5/2023

Programming for Civil Engineers

5

5

Repeat

- Calculate loads on each slab
- Draw moment and force diagrams
- Design:
 - Slabs
 - Beams
 - Columns
 - Foundation
- For Each loop
- Each iteration:
 - Calculate
 - Draw
 - Design

10/5/2023

Programming for Civil Engineers

6

6

Numerical Integration

$$y = -3x^2 + 12$$

$$\int_0^1 (-3x^2 + 12) dx$$

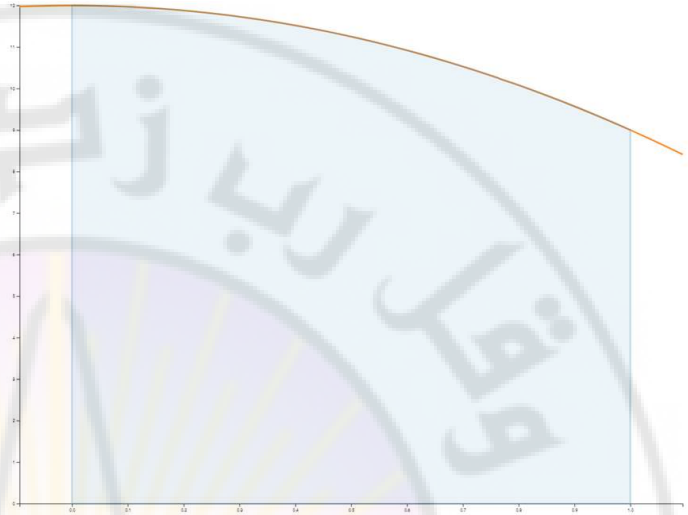
$$[-x^3 + 12x]_0^1 = 11 - 0 = 11$$

Rectangles:

1. $\Delta x = 1$ $x = 0.5$ $y = 11.25$
2. 11.06
5. 11.01

For loop, defined number of iterations

Iteration/rectangle Δx x y $area$



10/5/2023

Programming for Civil Engineers

7

7

Prime Numbers Example

To find all prime numbers less than or equal to integer n (Eratosthenes' method)

1. Create a list of consecutive integers from two to n : (2, 3, ..., n).
2. Initially, let p equal 2, the first prime number.
3. Strike from the list all multiples of p less than or equal to n . ($2p$, $3p$, $4p$, etc.)
4. Find the first number remaining on the list after p (this number is the next prime); replace p with this number.
5. Repeat steps 3 and 4 until p^2 is greater than n .
6. All the remaining numbers in the list are prime.

10/5/2023

Programming for Civil Engineers

8

8

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

10/5/2023

Programming for Civil Engineers

9

9

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Prime numbers

2	3	5	7
11	13	17	19
23	29	31	37
41	43	47	53
59	61	67	71
73	79	83	89
97	101	103	107
109	113		

10/5/2023

Programming for Civil Engineers

10

10

Loops

- Run line(s) of code repetitively
- Known number of iterations:
 - Items in a collection, e.g. calculate volume of all beams
 - Using index with start and end values, e.g. numerical integration
- Known condition
 - Unknown number of iterations, e.g. select section for steel member

10/5/2023

Programming for Civil Engineers

13

13

Loops

- for each element in a collection

```
For Each number In numbers
    WriteLine(number)
Next
```

- a specified number of times

```
For number = 1 To 4
    WriteLine(number)
Next
```

- until a condition is False (or True)

```
While number <= 4
    WriteLine(number)
End While
```

10/5/2023

Programming for Civil Engineers

14

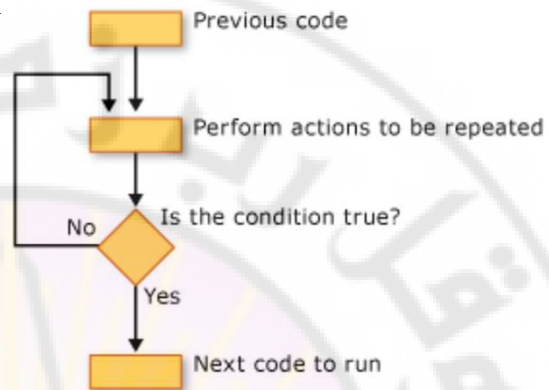
14

Loops

```
Dim numbers() As Integer = {1, 2, 3, 4}
For Each number1 In numbers
    WriteLine(number1)
Next

For number2 = 1 To 4
    WriteLine(number2)
Next

Dim number3 As Integer = 1
While number3 <= 4
    WriteLine(number3)
    number3 = number3 + 1
End While
```



10/5/2023

Programming for Civil Engineers

15

15

For Each

```
Dim numbers() As Integer = {1, 2, 3, 4}

For Each number In numbers
    WriteLine(number)
Next
```

10/5/2023

Programming for Civil Engineers

16

16

For Each Syntax

```
For Each item In items
    ' statements
Next
```

10/5/2023

Programming for Civil Engineers

17

17

For

```
For number = 1 To 10
    WriteLine(number)
Next
For number = 1 To 10 Step 3
    WriteLine(number)
Next
For number = 2 To 10 Step 3
    WriteLine(number)
Next
For number = 1 To 10 Step -2
    WriteLine(number)
Next
For number = 10 To 1 Step -2
    WriteLine(number)
Next
```

1 2 3 4 5 6 7 8 9 10

1 4 7 10

2 5 8

10 8 6 4 2

10/5/2023

Programming for Civil Engineers

18

18

For Syntax

Index i Or use start, end and step 1st Could be last
For counter [**As** datatype] = start **To** end [**Step** step]
 ' statements Default = 1
Next [counter]
 unlike ' counter

10/5/2023

Programming for Civil Engineers

19

19

While

```

code1
While number <= 4
  code2
  number = number + 1
End While
  
```

- Use condition instead of counter
- Unknown number of iterations
- Endless loop
- code1 : Needs to satisfy the condition to get in
- code2 : Needs to break the condition to get out

10/5/2023

Programming for Civil Engineers

20

20

While Endless Loop

```
While True
  If number > 10 Then
    Exit While
  End If
End While

While True
  Dim input As Integer
  input = ReadLine()
  If input = 0 Then
    Exit While
  End If
End While
```

10/5/2023

Programming for Civil Engineers

21

21

Loop Exit / Continue

```
While condition
  ' statements
  Continue While ' jump back to While
  ' statements
  Exit While ' jump to End While
  ' statements
End While
```

- GoTo Not recommended
- Continue For / Exit For
- Continue Do / Exit Do

10/5/2023

Programming for Civil Engineers

22

22

Do

```
While condition
  ' statements
End While
```

```
Do
  ' statements
Loop While condition

Do While condition
  ' statements
Loop
```

```
Do
  ' statements
Loop Until condition

Do Until condition
  ' statements
Loop
```

- Check condition at the beginning or end of the iteration
- Check for True or False
 - **While** condition
 - **Until Not** condition

10/5/2023

Programming for Civil Engineers

23

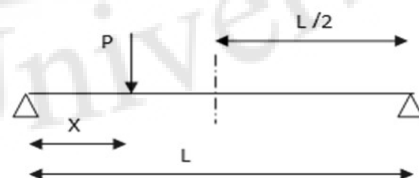
23

Input Validation

```
Dim L, X As Double

WriteLine("Enter the length of the beam [m] ")
L = ReadLine()
WriteLine("Enter the load distance from the left [m] ")
X = ReadLine()

If (L <= 0 Or X < 0 Or X > L) Then
  WriteLine("Input Error!")
Else
  ' Calculate...
End If
```



10/5/2023

Programming for Civil Engineers

24

24

Input Validation

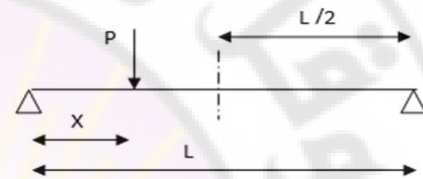
```

Dim L, X As Double

Do
    WriteLine("Enter the length of the beam [m] ")
    L = ReadLine()
Loop While (L <= 0)

Do
    WriteLine("Enter the load distance from the left [m] ")
    X = ReadLine()
Loop While (X < 0 Or X > L)

```



10/5/2023

Programming for Civil Engineers

25

25

Input Validation

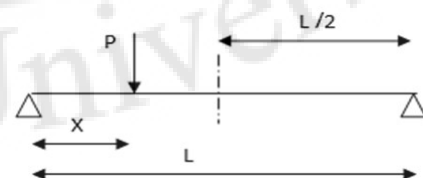
```

Dim L, X As Double

Do
    WriteLine("Enter the length of the beam [m] ")
    L = ReadLine()
Loop Until (L > 0)

Do
    WriteLine("Enter the load distance from the left [m] ")
    X = ReadLine()
Loop Until (X >= 0 And X <= L)

```



10/5/2023

Programming for Civil Engineers

26

26

Loops Syntax

```

For counter = start To end [ Step step ]
    ' statements
Next [ counter ]

While condition
    ' statements
End While

Do [ { While | Until } condition ]
    ' statements
Loop

For Each item In items
    ' statements
Next

Do
    ' statements
Loop [ { While | Until } condition ]

```

10/5/2023

Programming for Civil Engineers

27

27

Nested Loops

	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
<u>1</u>	1,1	1,2	1,3	1,4
<u>2</u>	2,1	2,2	2,3	2,4
<u>3</u>	3,1	3,2	3,3	3,4

```

For row = 1 To 3
    For column = 1 To 4
        WriteLine("Cell (" & row & ", " & column & ")")
    Next column
Next row

```

```

Cell (1, 1)
Cell (1, 2)
Cell (1, 3)
Cell (1, 4)
Cell (2, 1)
Cell (2, 2)
Cell (2, 3)
Cell (2, 4)
Cell (3, 1)
Cell (3, 2)
Cell (3, 3)
Cell (3, 4)

```

10/5/2023

Programming for Civil Engineers

28

28

Ends

- If / End If : Same number
- For / Next, While / End While, Do / Loop: Same number

$$\bullet (1+2 \times (3+4)) = (1+2 \times (3+4))$$

$$\neq (1+2 \times (3+4))$$

10/5/2023

Programming for Civil Engineers

29

29

Examples: Sum Odd Numbers

```

Dim a, b As Integer
WriteLine("Enter a then b (a<=b): ")
a = ReadLine()
b = ReadLine()
If (a > b) Then
    WriteLine("Wrong a > b")
Else
    If (a Mod 2 = 0) Then
        a = a + 1
    End If
    Dim sum As Integer = 0
    For i = a To b Step 2
        sum = sum + i
    Next
    WriteLine("Sum = " & sum)
End If

```

اكتب برنامج بلغة VB.Net
لحساب مجموع الأعداد الفردية
الواقعة بين عددين A و B
حيث $A \leq B$

30

Examples: Factorial

```
Dim n As Integer
WriteLine("Enter n (n>=0): ")
n = ReadLine()
If (n < 0) Then
    WriteLine("Error, no factorial")
Else
    Dim P As Long = 1
    For i = 1 To n
        P = P * i
    Next
    WriteLine("Factorial = " & P)
End If
```

اكتب برنامج بلغة VB.Net لحساب
عاملي عدد صحيح
(مع مراعاة جميع الحالات الممكنة)

- Use long
- For product start at 1, for sum start at 0
- $0! = 1$
- For i = 1 To n
- For i = n To 1 Step -1

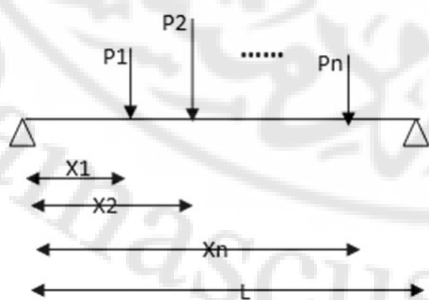
10/5/2023

Programming for Civil Engineers

31

31

Example: Calculate Reactions



اكتب برنامج بلغة VB.Net لحساب
ردّي الفعل عند المسندين لجائز
معرض لعدد من الحمولات المركزة:

10/5/2023

Programming for Civil Engineers

32

32

Example: Calculate Reactions

```

WriteLine("Enter beam length: ")
Dim L As Double = ReadLine()

WriteLine("Enter number of loads: ")
Dim N As Integer = ReadLine()

If (L <= 0 Or N < 0) Then
    WriteLine("Wrong Input!")
Else
    End If

```

10/5/2023

Programming for Civil Engineers

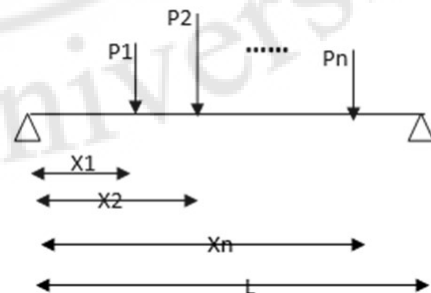
33

33

```

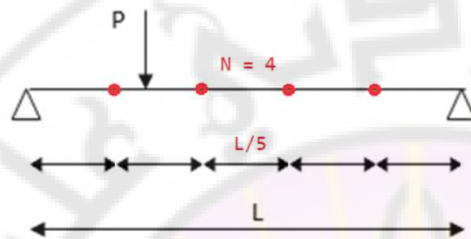
Dim P, X, SigmaM, SigmaY, RR, RL As Double
SigmaM = 0
SigmaY = 0
For i = 1 To N
    WriteLine("Enter P then X for Load " & i)
    P = ReadLine()
    X = ReadLine()
    If (X < 0 Or X > L) Then
        WriteLine("Wrong entry, it will be ignored!")
    Else
        SigmaY = SigmaY + P
        SigmaM = SigmaM + P * X
    End If
Next
RR = SigmaM / L
RL = SigmaY - RR
WriteLine("RR = " & RR)
WriteLine("RL = " & RL)

```



34

Example: Calculate Bending Moment



اكتب برنامج بلغة VB.Net لحساب عزم الانعطاف في عدد من النقاط الموزعة N على طول الجائز تحت تأثير حمولة مركزة و طباعة النتائج في جدول:

10/5/2023

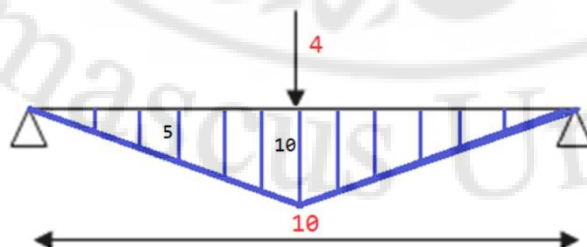
Programming for Civil Engineers

35

35

Example: Calculate Bending Moment

```
Enter P (t):
4
Enter L (m):
10
Enter X (m):
5
Enter N:
11
X      M
----  ---
0      0
0.83  1.67
1.67  3.33
2.5    5
3.33  6.67
4.17  8.33
5      10
5.83  8.33
6.67  6.67
7.5    5
8.33  3.33
9.17  1.67
10     0
```



اكتب برنامج بلغة VB.Net لحساب عزم الانعطاف في عدد من النقاط الموزعة N على طول الجائز تحت تأثير حمولة مركزة و طباعة النتائج في جدول:

10/5/2023

Programming for Civil Engineers

36

36

```

Dim L, P, X As Double
WriteLine("Enter P (t):")
P = ReadLine()
WriteLine("Enter L (m):")
L = ReadLine()
WriteLine("Enter X (m):")
X = ReadLine()

WriteLine("Enter N:")
Dim N As Integer = ReadLine()

Dim RL, RR, M, Delta As Double
RR = P * X / L
RL = P - RR
Delta = L / (N + 1)

WriteLine("X" & Chr(9) & "M")
WriteLine("---" & Chr(9) & "---")
For D = 0 To L Step Delta
    If (D <= X) Then
        M = RL * D
    Else
        M = RR * (L - D)
    End If
    WriteLine(Round(D, 2) & Chr(9) & Round(M, 2))
Next

```

10/5/2023

Programming for Civil Engineers

37

37

Example: Sum 1

```

Dim N, sum As Integer
WriteLine("Enter a number: ")
N = ReadLine()
sum = 0
Do While (N >= 0)
    sum = sum + N
    WriteLine("Enter a number: ")
    N = ReadLine()
Loop
WriteLine("Sum: " & sum)

```

اكتب برنامج بلغة VB.Net يقوم بحساب مجموعة من الأعداد الموجبة عددها غير معروف. يستمر البرنامج بالعمل وجمع الأعداد طالما القيم المدخلة موجبة، الى أن يتم ادخال عدد سالب أو صفر فيطبع البرنامج نتيجة الجمع ويتوقف عن العمل.

10/5/2023

Programming for Civil Engineers

38

38

Example: Sum 2

```
Dim N, sum As Integer
Do
    sum = sum + N
    WriteLine("Enter a number: ")
    N = ReadLine()
Loop While (N > 0)
WriteLine("Sum: " & sum)
```

اكتب برنامج بلغة VB.Net يقوم بحساب مجموعة من الأعداد الموجبة عددها غير معروف. يستمر البرنامج بالعمل وجمع الأعداد طالما القيم المدخلة موجبة، الى أن يتم ادخال عدد سالب أو صفر فيطبع البرنامج نتيجة الجمع ويتوقف عن العمل.

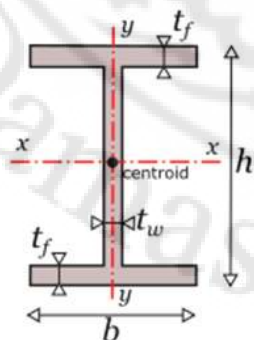
10/5/2023

Programming for Civil Engineers

39

39

Example: Section



اكتب برنامج بلغة VB.Net يقوم بقراءة المساحة الصغرى والمساحة العظمى التي يجب أن يحققها مقطع جاذز I-Beam ومن ثم يقوم بقراءة بيانات مقاطع من نوع I (ارتفاعه h وعرض الجناح b ونفس السماكة للجناح والجسد t) الى أن يصل الى مقطع تحقق مساحته الشروط المطلوبة

10/5/2023

Programming for Civil Engineers

40

40

```

Dim Amin, Amax, area, t, h, b As Double
WriteLine("Enter Minimum then Maximum Section Area in cm2: ")
Amin = ReadLine()
Amax = ReadLine()
Do
    WriteLine("Try a section: ")
    WriteLine("Enter Beam Section Width in mm: ")
    b = ReadLine()
    WriteLine("Enter Beam Section Height in mm: ")
    h = ReadLine()
    WriteLine("Enter Beam Section Thickness in mm: ")
    t = ReadLine()
    area = (2 * b * t + (h - 2 * t) * t) / 100
    WriteLine("Section Area= " & area & " cm2")
Loop Until (area >= Amin And area <= Amax)
WriteLine("Section is Acceptable")

```

10/5/2023

Programming for Civil Engineers

41

41

Questions



10/5/2023

Programming for Civil Engineers

42

42