① Conversions Of Number System

★ Binary (2)

(i) Denary to Binary:
Prime factorising denary no. w/ 2.

eg: $71_{(10)}$ =

2	71
2	35 – 1
2	17 – 1
2	8 – 1
2	4 – 0
2	2 – 0
	1 – 0

$= 1001_{(2)}$

(iii) Hexadecimal to Binary:
Separately prime factorising each Hexadecimal digit w/ 2. Following the order as in Hexadecimal no.

eg: 9A

2	9
2	4 – 1
2	2 – 0
	1 – 0

2	10
2	5 – 0
2	2 – 1
	1 – 0

$= 1001\ 1010_{(2)}$

★ Denary (10)

(i) Binary to Denary:
Multiply each binary digit w/ 2^power. Add all.

eg: $101_{(2)}$
$= 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1$
$= 4 + 0 + 1$
$= 5_{(10)}$

(ii) Hexadecimal to Denary:
Multiply each hexadecimal digit w/ 16^power. Add all.

eg: $2B_{(16)}$
$= 16^1 \times 2 + 16^0 \times 11$
$= 32 + 11$
$= 43_{(10)}$

★ Hexadecimal (16)

(i) Binary to Hexadecimal:
Make groups of four digits in binary no. Add "0" if required to make upto four on the left side of no. Add each group no. a/ multiplying w/ 2^power. A/ addition, each group is a hexadecimal digit.

eg: (Added) 0000 1010 0100 $_{(2)}$
a) $2^3 \times 0 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 0$
$= 0$
b) $2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 0$
$= 14$
c) $2^3 \times 0 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 0$
$= 9$
$\therefore 0\ 14\ 9_{(16)} \approx 0E9_{(16)}$

(ii) Denary to Hexadecimal:
Prime factorising denary no. w/ 16.

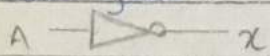
eg: $268_{(10)}$ =

16	268
	16 – 12
	1 – 0

$= 10\ 12_{(16)} \approx 10C_{(16)}$

② Logic Gates

* Not gate:



→ Logic = NOT A

→ Boolean = $\bar{a}$

A	x
0	1
1	0

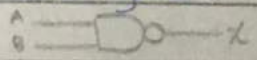
* And gate:

→ Logic = A AND B

→ Boolean = $a \cdot b$

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

* Nand gate: (NOT AND)



→ Logic = A NAND B

→ Boolean = $\overline{a \cdot b}$

A	B	x
0	0	1
0	1	1
1	0	1
1	1	0

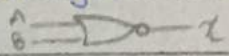
* Or gate:

→ Logic = A OR B

→ Boolean = $a + b$

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

* Nor gate: (NOT OR)

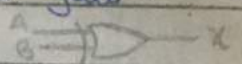


→ Logic = A NOR B

→ Boolean = $\overline{a+b}$

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

* Xor gate:



→ Logic = A XOR B

→ Boolean = $(a \cdot \bar{b}) + (\bar{a} \cdot b)$

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

③ Visual Basic Programming

Introduction:

* Function: Collection of instructions.
* Program: Collection of functions.
* Software: Collection of programs.
* Programming language.
* High-level language.
* Easy to learn.

Syntax:

* Sub Main()
  Working
  End Sub
* User-defined term;
  Start with: Letter, 1-9, _.
* Comments: ①Comment xyz②.
* = : equal , <> : not equal.
* Console.Readkey : To hold a program.

Datatypes: "Dim variable As _____"

1) Boolean: True / False
2) Char: Single letter / no.
3) String: Any sentence / no.
4) Single: No. only, less values, holds decimals
5) Double: No. only, more values, holds decimals
6) Integer: No. only, no decimal but -ive values too.

② Conditional Statements:

Set of rules performed if a certain condition is met.

a) If statement;

(i) If Then:

Syn
```
If condition Then
    statement
End if
```

Eg
```
If a > b Then
    c = a
End if
```

(ii) If Then Else:

Syn
```
If condition Then
    statement
Else
    other statement
End if
```

Eg
```
If a>b Then
    c=a
Else
    d=a
Endif
```

b) Case structure ;

Syn
```
Select Case variableName
    Case condition
    statement
    Case condition
    statement
    Case Else statement
End Select
```

Eg
```
game = console.readline()
Select Case game
    Case Is = "GTA 1"
    console.writeline("1")
    Case Is = "GTA 2"
    console.writeline("2")
    Case Else
    console.writeline("No")
End Select
```

Eg
```
numb = console.readline()
Select Case numb
    Case Is > 0
    console.writeline("G")
    Case Is < 0
    console.writeline("L")
    Case Else
    console.writeline("Null")
End Select
```

➡ ③ Loops :

Set of instructions continuously performed until a certain condition is met.

a) For ... To ... Next ;

Syn
```
For condition To condition
    statement
Next
```

Eg
```
For age = 1 To 10
    age = console.readline()
Next
```

b) While ... End While ;

Condition tested at start of loop.

Syn
```
While condition
    statement
    increment   (To prevent infinite loop)
End While
```

Eg
```
x = 3
While  x = 3
    console.writeline("3")
    x += 1
End While
```

c) Do loop ;

Condition tested after first cycle.

(i) Do... While (~~While~~ the condition...)

Syn
```
Do
    statement
    increment
Loop While condition
```

Eg
```
x = 0
Do
    console.writeline("x")
    x += 1
Loop While x > 5
        Output: x
```

(ii) Do... Until (Till the condition...)

Syn
```
Do
    statement
    increment
Loop Until condition
```

Eg
```
x = 11
Do
    console.writeline("x")
    x += 1
Loop Until x = 13
        Output: x, x
```

Arrays :

An array can store multiple values.

Syn Dim arrayName (no. of values) As Datatype

Eg Dim marks (10) As Integer

* Filling an Array;

```
Dim marks (10) As Integer
Dim c As Integer
For c = 1 To 10
    marks(c) = console.readline
Next
```

➡ ⑥ Functions:

Used to perform a specific task. Can be called in a program where required. Returns a value after execution.

(i) ② Numeric Functions:

Contains only numbers & are suitable for numeric calculations.

a) Int function;

Only returns the integer value without rounding.

Syn

```
Dim varName As Datatype
varName = console.readline
console.writeline (Int (varName))
```

Eg

```
Dim x As Single
x = 1.009
console.writeline (Int (x))
```

Output: 1

b) Round function;

Rounds the value to nearest integer.

Syn

```
Dim varName As Datatype
varName = console.readline
console.writeline (Math.Round (varName))
```

Eg
```
Dim x As Single
x = 1.009
console.writeline (Math.Round(x))
```
Output : 1

(iii) (4) String Functions :

May contain numbers and/or strings but not suitable for numeric calculations.

a) Left function ;

Gives characters from left side i.e abcd, 2 = ab.

Syn
```
Dim varName As Datatype
varName = console.readline
console.writeline (Left (varName, No. of characters))
```

Eg
```
Dim x As String = "abdullah"
console.writeline (Left (x, 3))
```
Output : abd

b) Right function ;

Gives characters from right side i.e abcd, 2 = cd.

Syn
```
Dim varName As Datatype
varName = console.readline
console.writeline (Right (varName, No. of characters))
```

Eg
```
Dim x As String = "abdullah"
console.writeline (Right (x, 3))
```
Output : lah

c) Len function ;

Returns the length / no. of characters of variable.

Syn
```
Dim varName As Datatype
varName = console.readline
console.writeline (Len (varName))
```

Eg
```
Dim x As String = "abdullah"
console.writeline (Len(x))
```
Output: 8

d) Mid function;
Returns the required number of characters from specified location.

Syn
```
Dim varName As Datatype
varName = console.readline
console.writeline (Mid (varName, Character's starting position,
                        No. of characters from starting position))
```

Eg
```
Dim x As String = "abdullah"
console.writeline (Mid (x, 3, 3))
```
Output: dul

④ Extras

★ Pseudocode Programming:

a) Conditional Statements;
- If ... Then ... Else ... Endif
- Case ... Of ... Otherwise ... End case

b) Loops;
- For ... To ... Next
- Repeat ... Until
- While ... Do ... Endwhile