Module-1.

(I). A 'program' in general, is a sequence of instructions written so that a computer can perform a certain task.

A 'script' is code written in scripting language. A scripting language is nothing but a type of programming language in which we can write code to construct another software application.

1) ~~A script A pro~~

2) A typical digital computer system has four basic functional elements:
(1) input-output equipment
(2) main memory
(3) control unit
(4) arithmetic-logic unit.

3) Compiler →(1) Compiler scans the whole program in one go. (2) As it scans the code in one go, the errors (if any) are shown at the end together.
(3) Main advantage of compilers is its execution time.
(4) It converts the source code into object code.
(5) It does not require source code for later execution.

Interpreter →(i) Translates programs one statement at a time.
(ii) Considering it scans code one line at a time, errors are shown line by line.
(iii) Due to interpreters being slow in executing the object code, it is preferred less.
(iv) It does not convert source code into object code instead it scans line by line.
(v) It requires the source code for later execution.

4) RAM is volatile memory used to hold instructions and data of currently running programs. It loses integrity after loss of power. ~~RAM memory modules are in.~~
[Volatile memory is computer memory that requires power to maintain the stored information; it retains its contents while powered on but when the

power is interrupted, the stored data is quickly lost]

5) An operating system plays the following roles:
(i) Controls the backing store and peripherals such as ~~scanners~~ scanners and printers.

(ii) Deals with the transfer of programs in and out of memory.
(iii) Organises the use of memory between programs.
(iv) Organises processing time between programs and users.
(v) Maintains security and access rights of users.

6) Types of memories:-

(i) CPU register. (ii) Cache ~~memory~~ memory
(iii) Primary / Main memory. (iv) Secondary Memory / Mass storage.

Primary memory.	Secondary memory
1) Primary memory is temporary.	Secondary memory is permanent.
2) Directly accessibly by processor (CPU).	Not directly accessible by CPU.
3) Nature of parts of primary memory varies, RAM - volatile in nature.	Non-volatile in nature.
4) The memory devices used for primary memory are seconductor memories.	The ~~sec~~ secondary memories are magnetic and optical memories.

Cache memory, also called cache, supplementary memory system that stores frequently used instructions and data for quicker processing by the C.P.U of a computer.

2022/2/28 22:03

Registers memory is the smallest and fastest memory in a computer. It is used to quickly accept, store and transfer data and instruction that are being used immediately by the CPU.

Module-7:

1.) ~~5/2 → given is used for floor division. 5/2 It gives~~

2)

Relational operators	Logical operators.
These operators are used to compare the value of two variables.	These operators are used to perform logical operations on the given two variables.

1) 5/2 → The / operator is used to assign quotient. Thus, 5/2 gives the result as 2.5

5//2 → The // operator is used to assign floor division. 5//2 gives 2 [∵ 2 is the closest integer to 2.5].

Module-8:

1.) The ~~ma~~ main difference between break and ~~conte~~ continue statement is that when break keyword is encountered, it will exit the loop.

In case of continue keyword, the current iteration ~~that~~ that is running will be stopped, and it will proceed with the next iteration.

Page

The pass statement is used as a placeholder inside loops, functions, class, if-statement that is meant to be implemented later. It is a null statement.

2) Deep copy → It is a process in which the copying process occurs recursively. It means first constructing a new collection object and then populating it with copies of the child objects found in the original. In case of deep copy, a copy of object is copied in another object. Any changes made to a copy of the object do not reflect in the original object.

Shallow copy → A shallow copy means constructing a new collection object and then populating it with the references to the child objects found in the original. The copying process does not recurse and therefore wont create duplicates. Any changes made to a copy of object do ~~not~~ reflect in the original function.

deepcopy is implemented using "deepcopy()" function.
shallow copy is implemented using "copy()" function.

3) The else keyword in a for loop specifies a block of code to be executed when the loop is finished. The else block just after for loop is executed only when the loop is not terminated by a break statement.

Example:

```
for i in range(1,4):
    print(i)
else:
    print("No Break")
```

2022/2/28 22:03

Output:

1

2

3

No Break

4)

While loop	for loop.
i) Used when the user does not know about the number of iterations needed.	Used when the users knows about the no. of iterations needed.
~~Syntax~~ ~~[illegible]~~ The loop variable must be incremented before the loop. If the user fails to increment the loop variable, then the loop will not terminate.	In for loop, variable is initialised, checked and incremented in this specific order. All these processes are performed inside this loop.

4) For loop → allows a programmer to execute a sequence of statements several times, it abbreviates the code which helps to manage loop variables.

While loop → allows a programmer to repeat a single statement or a group of statements for the TRUE condition. It verifies the condition before executing the loop.

For loop:

```
for i in range(11):
    print(i)
```

while loop:

```
i=0
while (i<=10):
    print(i)
    i=i+1
```

Module-7).

1.) / operator → performs division.
Divides left hand operator by right hand operator. 5/2 gives 2.5 as the result.

// operator → performs floor division.
the division of operands where the results the quotient in which the digital after the decimal point are removed but if one of the operands is negative the result is floored (rounded away from 0).
5//2 gives 2 ~~by~~ as the result.

2) Relational operators are used to compare the values of two expressions. ~~Re~~ They are binary operators because they require two operands to operate.

Logical operators are typically used with Boolean values. These are used to ~~perform~~ perform logical operations on the given two variables.

Module-5.

1) Algorithm → A process or set of rules to be followed in calculation or other problem-solving operations." Thus algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed upon in order to get the expected results.

2022/2/28 22:04

Page

Example: Algorithm of linear search.

1.) Start from the leftmost element of arr[] and one by one compare x with each element of arr[].

2.) If x matches with an element, return the index.

3) If x does not match with any of the elements, return -1.

Flowchart:

A flowchart is a graphical representation of an algorithm. ~~Programmers~~ It is often used ~~as~~ as a programming-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of ~~...~~ information and processing. (give an example).

A program is a detailed plan or procedure for solving a problem with a computer, ~~more~~ more specifically, an unambiguous, ordered sequence of computational instructions necessary to achieve such a solution.

OR

A sequence of instructions in a programming language that a computer can execute ~~or~~ or interpret.

2) Algorithm definition.

Properties of algorithm:

(i) Non-ambiguity → Each step in an algorithm should be non-ambiguous. That means each instruction should be clear ~~and~~ and precise. The instruction in any algorithm should not denote any conflicting meaning. This property also indicates the effectiveness of algorithm.

2) Range of input → should be specified. This is because normally the algorithm is input driven and if the range of input is not being specified, then ~~algor~~ algorithm can go in an infinite state.

(3) Multiplicity → The same algorithm can be represented into several different ways. This means ~~we can write in simple English~~ the sequence of instruction can be written in the form of a pseudo code.

(4) Speed → the algorithms written ~~on~~ using some specified ideas. But such algorithms should be efficient and should produced the output with fast speed.

(5) Finiteness → The algorithm ~~sh~~ ~~should~~ should be finite. That ~~means~~ means after performing required operators it should be terminate.

3) Time complexity is the amount of time taken by an algorithm to run, as a function of the length of the input. It measures the time taken to execute each statement of code in an algorithm.

Space complexity of an alogorithm is the amount of memory space required to solve an instance of the computational problem as a function of characteristics of the input.

2022/2/28 22:04

4) ~~B~~ Best case → No. of computations is minimum.

Average case → No. of computations is average.

Worst case → No. of computations is maximum.

For example, in modified bubble sort,

[15, 20, 25, 40, 45]

No. of comparisons = 4

Time = 4 × C_1 (Best case).

[45, 40, 25, 20, 15].

No. of comparisons → 4
3
2
1

Total no. of comparisons = 10

Time = 10 × C_1 (Worst case)

[10, 15, 45, 20, 25].

No. of comparisons = 7

Time = 7 × C_1 (Average case)

5) The limiting behaviour of the execution time of an ~~algo~~ algorithm when the size of the problem goes to infinity. This is usually denoted in big-O notation.

2022/2/28 22:04

Date __/__/__
Page ____

(1) Big Oh notation (O):

It is used to describe asymptotic upper bound.

Mathematically, if f(n) describes the running time of an algorithm; f(n) is O(g(n)) if there exist +ve constant C and n0 such that

$0 <= f(n) <= c\,g(n)$ for all $n >= n0$

n? used to give upper bound in a function.
If a function is O(n), it is automatically O(n-square) as well.

(2) Big Omega notation (Ω):

Just like O-notation provide an asymptotic upper bound, Ω notation provides asymptotic lower bound.

If f(n) be the running time of an algorithm.
f(n) is said to be Ω(g(n)) if there exists positive constant C and (n0) such that
$0 <= c\,g(n) <= f(n)$ for all $n >= n0$

n? used to given lower bound on a function.
If a function is Ω(n-square) it is automatically Ω(n) as well.

(3) Big Theta notation (Θ):-

Let f(n) define running time of an algorithm.
f(n) is said to be Θ(g(n)) is f(n) is O(g(n)) and f(n) is Ω(g(n))

Mathematically,
$0 <= f(n) <= c1 \cdot g(n)$ for $n >= n0$
$0 <= c2\, g(n) <= f(n)$ for $n >= n0$

2022/2/28 22:04

Date ___/___/___

Page ____

Merging both the equation, we get,

$0 <= C_2 g(n) \le f(n) <= C_1 g(n)$ for $n >= n_0$

~~6) 1) After the first iteration, the value of n will be 345 and~~

6) 1) Taking input to a number.
2) If the number is not equal to zero, divide the number using floor division operator.
3) ~~Decrement~~ Initialise a variable, say count, to 0 and increment it by one.
4) Print count to get the number of digits.

2022/2/28 22:04

Module- 3&4:

1) De-Morgan's First Theorem:

$$\overline{X+Y} = \bar{X}\bar{Y}$$

Proof using complementary laws:-

Complementarity laws state, $X+\bar{X}=1$ and $X.\bar{X}=0$

For the first part,

$$(X+Y)+(\bar{X}\bar{Y})=1$$

$$(X+Y)+\bar{X}\bar{Y} = ((X+Y)+\bar{X}).((X+Y)+\bar{Y}) \quad [\because X+YZ=(X+Y)(X+Z)]$$

$$=(X+\bar{X}+Y).(X+Y+\bar{Y})$$

$$=(1+Y).(X+1) \quad [\because X+\bar{X}=1]$$

$$=1.1 \quad [\because 1+X=1]$$

$$=1.$$

For the second part,

$$(X+Y).\bar{X}\bar{Y}=0$$

$$(X+Y).\bar{X}\bar{Y} = \bar{X}\bar{Y}.(X+Y) \quad [\because X(YZ)=(XY)Z]$$

$$=\cancel{\bar{X}\bar{Y}}$$

$$=\bar{X}\bar{Y}X+\bar{X}\bar{Y}Y \quad [\because X(Y+Z)=XY+XZ]$$

$$=X\bar{X}\bar{Y}+\bar{X}Y\bar{Y}$$

$$=0.\bar{Y}+\bar{X}.0 \quad [X.\bar{X}=0]$$

$$=0+0=0$$

Thus, $\overline{X+Y}=\bar{X}\bar{Y}$

De-Morgan's second Theorem:

$$\overline{X \cdot Y} = \bar{X} + \bar{Y}$$

Proof:

First part—

$LHS = XY + (\bar{X} + \bar{Y})$

$= (\bar{X} + \bar{Y}) + XY$ $\quad [\because X + Y = Y + X]$

$= (\bar{X} + \bar{Y} + X) \cdot (\bar{X} + \bar{Y} + Y)$ $\quad [\because X + YZ = (X+Y)(X+Z)]$

$= (X + \bar{X} + \bar{Y}) \cdot (\bar{X} + Y + \bar{Y})$

$= (1 + \bar{Y}) \cdot (\bar{X} + 1)$ $\quad [\because X + \bar{X} = 1]$

$= 1 \cdot 1$

$= 1 = RHS$

Second part:—

$XY \cdot (\bar{X} + \bar{Y}) = 0$

$LHS = XY \cdot (\bar{X} + \bar{Y})$

$= XY\bar{X} + XY\bar{Y}$ $\quad [\because X(Y+Z) = XY + XZ]$

$= X\bar{X}Y + XY\bar{Y}$

$= 0 \cdot Y + X \cdot 0$ $\quad [\because X \cdot \bar{X} = 0]$

$= 0 + 0$

$= 0 = RHS$

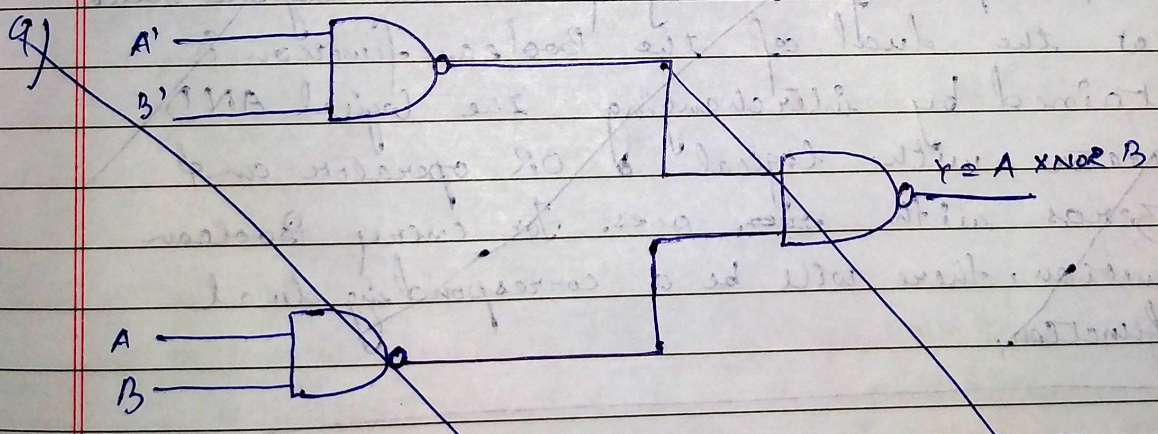
$XY \cdot (\bar{X} + \bar{Y}) = 0$

and $XY + (\bar{X} + \bar{Y}) = 1$

2) A universal gate is a gate which can implement any Boolean function without need to use any other gate type, e.g, NAND and NOR.

3) The NAND ~~and~~ and the NOR gates can be said to be universal gates since combinations of them can be used to accomplish any of the basic operations and can ~~be~~ thus ~~be~~ produce an Inverter, an OR gate or an AND gate. The non-inverting gates do not have this versatility since they can't produce an invert.

6) A truth table is a tabular representation of all the combinations of values for inputs and their corresponding outputs. It is a mathematical table that shows all possible outcomes that would occur from all the possible scenarios that are considered factual. Truth tables are usually used for logic problems as in Boolean algebra and electronic circuits.

9) 

XNOR Gate using only NAND Gate.

$Y = \overline{A \oplus B} = \overline{A}\cdot\overline{B} + A\cdot B$

5) $f = ab + bc + ca$

	a	b	c	ab	bc	ca	f
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	1	0	0	0	0	0
3	0	1	1	0	1	0	1
4	1	0	0	0	0	0	0
5	1	0	1	0	0	1	1
6	1	1	0	1	0	0	1
7	1	1	1	1	1	1	1

Min-term $= \Sigma\, 3, 5, 6, 7$
Max-term $= \Sigma\, 0, 1, 2, 4$

---

15) ~~The Principle of Duality of Boolean algebra states that the dual of the Boolean function is obtained by interchanging the logical AND operator with logical OR operator and zeros with ones. For every Boolean function, there will be a corresponding dual function.~~

---

14) Each property is grouped in pairs. Within each pair one statement can be obtained from the other statement by interchanging the OR and AND operators and by replacing the constants 0 and 1 by 1 and 0 respectively. This property is known as principle of duality.

2022/2/28 22:05

(5) Absorption law in Boolean algebra enables a reduction in a complicated expression to a simpler one by absorbing like terms.

1) $X + XY = X$

2) $X \cdot (X+Y) = X$

~~Alternative terms using principle of duality~~

Consensus theorem states that the consensus term of a disjunction is defined when the terms in function are reciprocals to each other (such as A and $\bar{A}$). Consensus theorem is defined in two statements (normal form and its dual).

1) $x.y + \bar{x}.z + y.z = x.y + \bar{x}.z$

2) $(x+y).(\bar{x}+z).(y+z) = (x+y).(\bar{x}+z)$

$$xy + x'z + yz$$
$$= xy.1 + \bar{x}z.1 + yz.1$$
$$= xy(z+\bar{z}) + \bar{x}z(y+\bar{y}) + yz(x+\bar{x})$$
$$= xyz + xy\bar{z} + \bar{x}\bar{y}z + xyz + \bar{x}yz$$
$$= xyz + xyz + xy\bar{z} + \bar{x}yz + \bar{x}yz + \bar{x}\bar{y}z$$
$$= xyz + xy\bar{z} + \bar{x}yz + \bar{x}\bar{y}z$$
$$= xy(z+\bar{z}) + \bar{x}z(y+\bar{y})$$
$$= xy.1 + \bar{x}z.1$$
$$= xy + \bar{x}z$$

2022/2/28 22:05

9) ~~XOR~~ XNOR USING NAND.

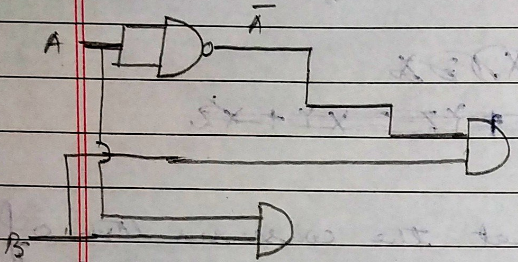
XNOR →

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

NAND →

A	B	Y
0	0	0
0	1	1
1	0	1
0	0	0

$\therefore XNOR = \bar{A}\cdot\bar{B} + AB = \overline{XOR}$



10) XOR using NAND.

A	B	Y
0	0	1
1	0	0
0	1	0
1	1	1

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$XNOR = \bar{A}\cdot\bar{B} + AB = \overline{XOR}$ $\therefore XOR = \bar{A}B + A\bar{B}$

$z = \bar{A}B + A\bar{B}$

$\bar{z} = \overline{\bar{A}B + A\bar{B}} = \overline{\bar{A}B}\cdot\overline{A\bar{B}} = (\bar{\bar{A}} + \bar{B})\cdot(\bar{A} + \bar{\bar{B}})$

$= (A + \bar{B})\cdot(\bar{A} + B)$

$= \underset{0}{A\cdot\bar{A}} + A\cdot B + \bar{A}\cdot\bar{B} + \underset{0}{\bar{B}\cdot B}$

$= AB + \bar{A}\bar{B}$

3(4→ 4, 7, 8, 13, 16, 17.

5→ 6, 7

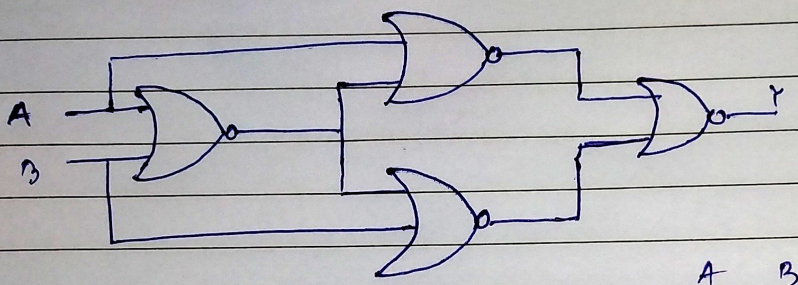
Date __/__/__

Page ____

(1) ~~XOR~~ XNOR using NOR.

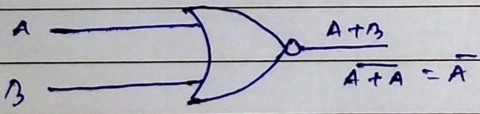
XNOR → $Y = (A \oplus B)'$

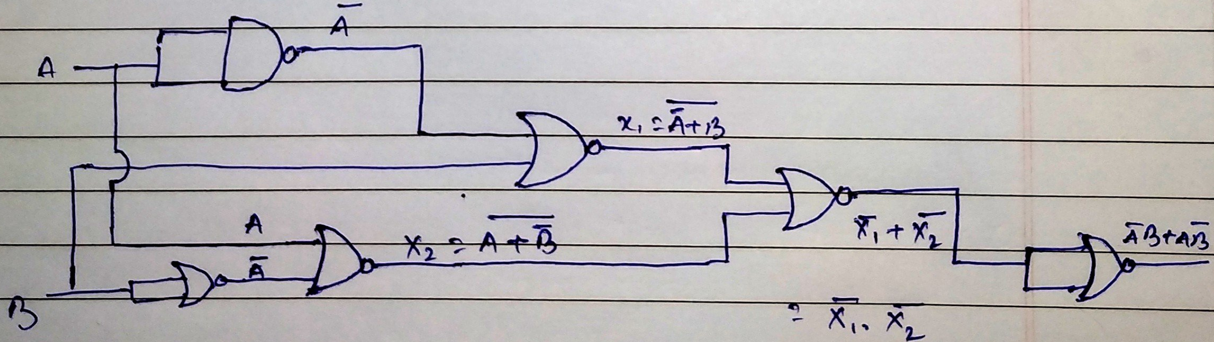
NOR → $Y = (A+B)'$



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

(2) XOR using NOR





$= \bar{x}_1 \cdot \bar{x}_2$

$= (\overline{\overline{\bar{A}+B}}) \cdot (\overline{\overline{A+\bar{B}}})$

$= (\bar{A}+B) \cdot (A+\bar{B})$

$= \underbrace{\bar{A} \cdot A}_{0} + \bar{A} \cdot \bar{B} + A \cdot B + \underbrace{B \cdot \bar{B}}_{0}$

$= \bar{A}\bar{B}$ ...