DAY-1

INTRODUCTION TO CSE

CSE → study of computational systems, algorithms & software design

It blends "Theory" & "mathematics" & "Engineering" to create practical systems that impact almost every aspect of modern life - from smartphones to banking healthcare, space research, and AI

CSE (other specs) → aims to equip students with knowledge and skills to solve complex real-world problems using computing principles and to innovate in areas such as cloud computing, IOT, ML, robotics & more

KEY AREAS OF COMPUTER SCIENCE:

AREA	DESCRIPTION
• Algorithms & Data Structures	Efficient methods to store, retrieve and manipulate data.
• Programming	writing code to instruct computers to perform tasks
• Computer Hardware & Architecture	understanding how computers physically operate
• Operating System	Managing hardware and software resources
• Databases	Systems to organize & access large amts. of data
• Computer Networks	How data is transferred across systems (e.g. internet)
• Software Engineering	Designing, Building, and maintaining reliable software
• Artificial Intelligence Engineering (AI)	Creating systems that simulate human intelligence.
• Cybersecurity	Protecting systems ~~that stm simulate~~ from digital attacks
• Human Computer Interaction (HCI)	Designing user friendly systems

⊛ Why study Computer Science

- Problem Solving: Build logical thinking and analytical skills
- Creativity: Design apps, games & AI systems
- Career Opportunity: High demand across industries
- Global Impact: Solve real-world challenges in health, climate, finance, etc.

⊛ What do Computer Science Engineers Do?

→ Develop mobile apps or enterprise software
→ Build websites and online platforms.
→ Design AI and ML models
→ Manage & secure networks
→ Conduct research in computing theory, various disciplines of computer science, AI, IOT, robotics etc.

FOUNDATION SKILLS NEEDED

- Logical Reasoning
- Basic Maths
- Interest in technology
- Problem solving mindset

WHAT IS ALGORITHM :

Step-by-step set of instructions used to solve a ~~and~~ pbm & perform a task

E.g. A recipe for making tea is an algorithm.

S-1 → Boil ~~water~~ water
S-2 → Add tea leaves
S-3 → Brew for 3 minutes
S-4 → Pour into cup
S-5 → Add milk/sugar as needed

Application of Algorithm

In Computer Sci, algo's are used to -

- Sort data (e.g. alphabetizing names)
- Search for items (e.g. finding contact in phone)
- Perform calculations (e.g. computing avg)
- Optimize process (e.g. routing in Google Maps)

Properties of good algorithm:

Input: Takes 0 or more inputs

Output: Produces atleast 1 output

Definiteness: Each step must be clearly defined

Finiteness: Must terminat after a finite no. of steps

Effectiveness: Each step must be basic enough to be performed

Representation of Algorithms:

- Natural Language: Descriptive steps in plain English
- Pseudo code: A lang-independent representation (keywords)
- Flow Charts: Diagramatic representation of logic

TYPES OF ALGORITHMS:

Type	Description	Example
Search Algorithm	Find an element in data	Linear Search, Binary Search
Sort Algorithm	Arranges data in order,	Bubble Sort, Merge Sort
Recursive Algorithm	Solves a pbm by solving smaller sub problems / instance of itself	Factorial, Fibonacci
Greedy Algorithm	Makes optimal choice at each step	Coin Change, Djikstra
Dynamic Problem.	Solves problems by combining solutions of sub problems	knapsack, Fibonacci
Back Tracking	Builds soln step by step, backtrack if needed	N-Queens, Sidoku
Dividing & Conquer	Breaks problem into smaller parts	Merge sort, Quick sort

Why Study Algorithm ?

- → Improves Problem-solving skills
- → Leads to efficient programs (fasts, less memory)
- → Helps in technical interviews and real world system design.

EXAMPLES OF BASIC ALGORITHMS

① Algorithm to Add two nos.

```
START
INPUT A & B
C = A+B
PRINT C
END
```

② Find maximum of Three Nums.

```
Step 1 : Start
step 2 : Read a, b, c
skp 3 : if a>b AND a>c:
            max = a
        else if b>c
            max = b
        else:
            max = c
step 4 : Print max
step 5 : stop
```

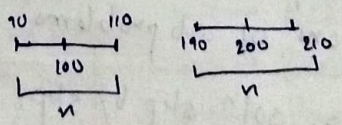
③ Check Even or Odd

```
S1:→ Start
S2:→ Read n
S-3:→ if n%.2 == 0,
        print "Even"
    Else
        print "Odd"
S-4:- Stop
```

④ Divisibity by 3 or 7

```
Start
Read +ve int 'n'
if (n%.3==0) OR (n%.7==0)
    return True
else
    return False
stop
```

⑤ check if a given no. is within 10 of 100 or 200




```
Start
Read the int n.
if |n-100| ≤ 10 OR |n-200| ≤ 10
    return True
else
    return False
STOP.
```

⑥ Sum of First n Nat. NOs.

```
Start
Read n
sum = 0
i = 1
while i ≤ n
    sum = sum + i
    i += 1
Print sum
stop
```

⑦ Algo to Compute Factorial

```
Start
Read n
fact = 1
i = 1
while i ≤ n
    fact = fact * i
    i += 1
Print fact
stop
```

EXERCISE

⊙ Write an algo to compute the sum of the two input values. If two input values are the same, the return triple their sum.

```
Start
Read a, b
sum = a + b
if a == b:
    ~~sum = a + b~~
    triplesum = 3 * sum
    Print (triple sum)
else:
    ~~str~~
    stop Print (sum)
```

⊙ Write an algo to read two given int and check if either of them is in range 100--200 inclusive

```
Start
Read a, b
if (a ≤ 200 & a ≥ 100) OR
   (b ≥ 100 & b ≤ 200):
    print 'True'
else
    print 'False'
stop
```

⊙ Product of first n natural numbers

```
Start
Read n
pdt = 1
i = 1
while i ≤ n
pdt = pdt * i
i += 1
print (pdt)
Stop
```

⊙ Find the Last digit of a given number

input: 1234

output: 4

```
Start
Read n
last = n % 10
print (Last)
stop
```