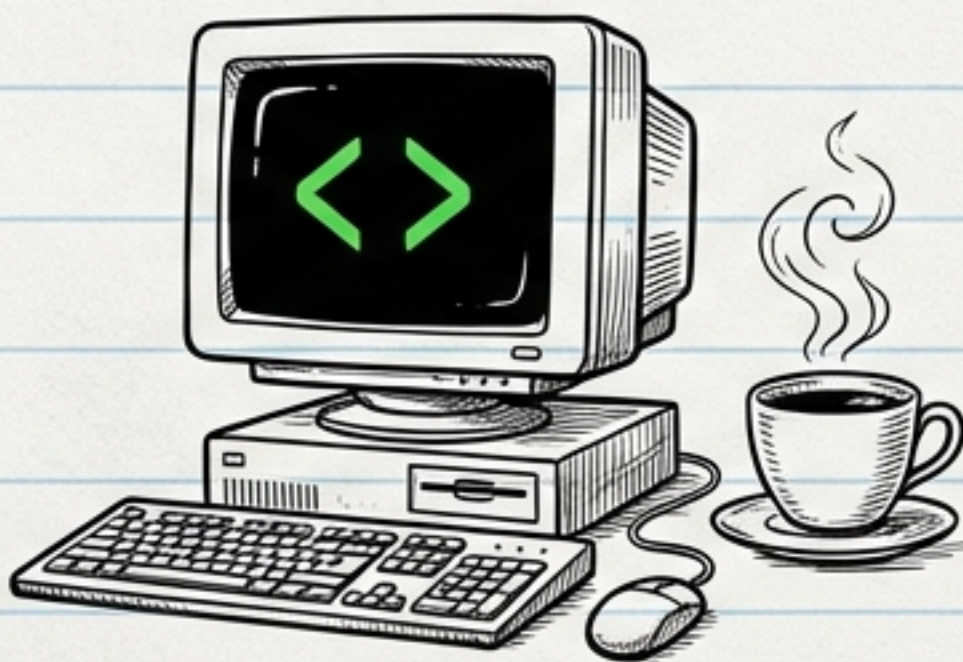


**TOP SECRET  
REVISION**

# Computer Awareness: Complete Revision Notes

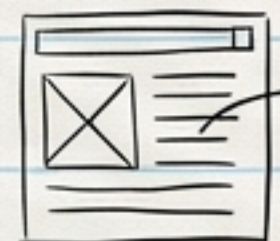
Handwritten Study Guide



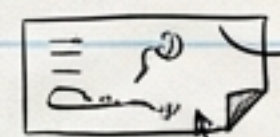
Visual Margin



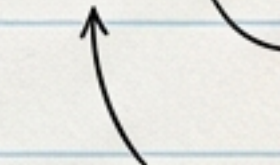
Hand diagrams



Example annotation



Indicated annotation notes



Meor diagrams

Visual Margin

## → Tabulating Machine (टेबुलेटिंग मशीन)

आविष्कारक- **हर्मन होलेरिथ** (1880 के दशक में)

इसका उपयोग वर्ष **1890** में अमेरिकी जनगणना (Census) में किया गया। होलेरिथ की टेबुलेटिंग मशीन कंपनी आगे चलकर वर्ष **1924** में इंटरनेशनल बिजनेस मशीन (**IBM**) बन गई।

## → पंचकार्ड (Punch Card)

कागज का बना एक कार्ड है जिसमें पंच द्वारा छेद बनाकर कम्प्यूटर डाटा तथा प्रोग्राम स्टोर किया जाता था।

यह **पहली इनपुट डिवाइस** है।

## → ABC- Atanasoff Berry Computer

आविष्कारक- **जॉन एटनासॉफ और क्लिफोर्ड बैरी** (1937-1942)

पहला **स्वचालित इलेक्ट्रॉनिक डिजिटल कम्प्यूटर**।



## → Mark 1 (मार्क 1)

वर्ष 1937 में, कम्प्यूटर के इतिहास में बड़े बदलाव शुरू हुए जब हॉवर्ड एकेन ने एक ऐसी मशीन विकसित करने की योजना बनाई जो बड़ी-बड़ी गणनाएं कर सके। वर्ष 1944 में आईबीएम (IBM) और हार्वर्ड के बीच साझेदारी के रूप में मार्क 1 कम्प्यूटर का निर्माण गया था। यह कम्प्यूटर जगत में एक नए करने वाला पहला प्रोग्राम योग्य डिजिटल कम्प्यूटर भी था। इसमें डाटा मैनुअल रूप से इंटर किया जाता था। इसमें स्टोरेज के लिए मैग्नेटिक ड्रम का प्रयोग किया जाता था।

## → ENIAC

ENIAC-Electronic Numerical Integrator And Computer (1945-46)

The first programmable, electronic general purpose digital computer.

इसमें 18000 वैक्यूम ट्यूब का उपयोग किया गया था।

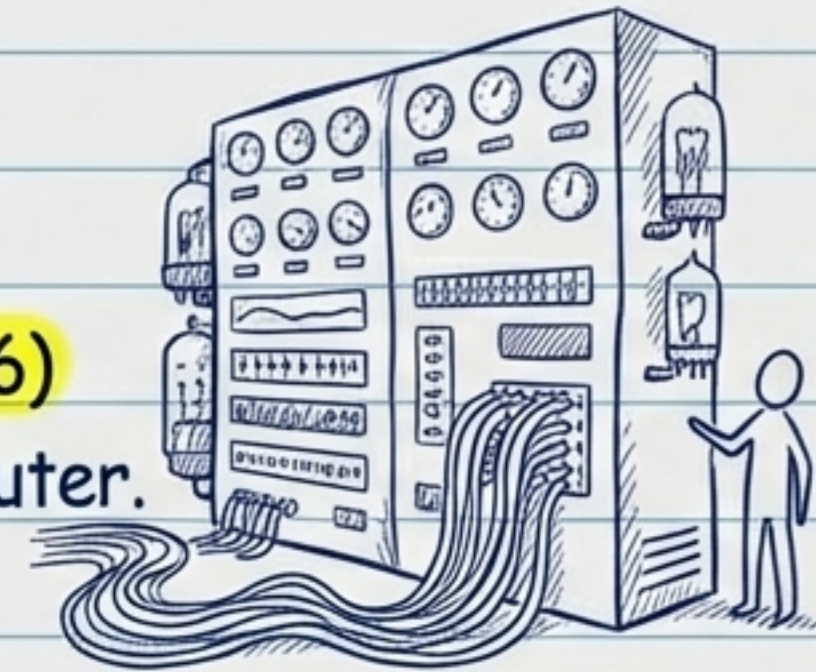
इसमें 20 Accumulators का संयोजन किया गया था।

## → EDSAC

EDSAC- Electronic Delay Storage Automatic Calculator (1947-49)

Second electronic digital stored program computer to go into regular service.

यह मर्करी डीलेय लाइन्स का प्रयोग मेमोरी के लिए और वैक्यूम ट्यूब का प्रयोग लॉजिक के लिए करता है।



EARLY COMPUTER  
(1940s)

## UNIVAC

UNIVAC - Universal Automatic Computer (1950-51)




आविष्कारक- प्रेस्पर एकर्ट और जॉन मोचली



यह इनपुट और आउटपुट के लिए मैग्नेटिक टेप का उपयोग करता है।

## अपोलो गाइडेंस कम्प्यूटर (AGC)

अपोलो गाइडेंस कम्प्यूटर नासा के अपोलो कार्यक्रम के लिए 1960 के दशक में विकसित की गई एक अत्याधुनिक तकनीक थी। यह पहला कम्प्यूटर था जिसने इंटीग्रेटेड सर्किट्स का उपयोग किया और चंद्रमा मिशनों के दौरान अंतरिक्ष यान के मार्गदर्शन, नेविगेशन और नियंत्रण में महत्वपूर्ण भूमिका निभाई। इस कम्प्यूटर का वजन लगभग 70 पाउंड (32 किग्रा) और आयतन लगभग 1 घन फुट था सहमा पाउंड (32 किग्रा) और आयतन लगभग 1 घन फुट था तथा इसको डिस्प्ले और कीबोर्ड इंटरफ़ेस का उपयोग करके संचालित किया जाता था। इसमें डिस्कीट सिलिकॉन इंटीग्रेटेड सर्किट चिप्स के साथ 2.048 मेगाहर्ट्ज की क्लॉक स्पीड वाला प्रोसेसर लगाया गया था। इस कम्प्यूटर में 2K शब्दों की RAM और 36K शब्दों की ROM (कोर रोप मेमोरी) के रूप में प्रयोग की गई। तथा यह कम्प्यूटर 64KB तक डेटा स्टोर कर सकता था। इस कम्प्यूटर का पावर कन्जप्शन लगभग 55 वाट था।



Generation	Hardware Components	Characteristics of Computer	Memory	Operating System	Speed	Example
प्रथम पीढ़ी (1942-1955)	Vacuum Tube 	निम्न स्तरीय प्रोग्रामिंग भाषा का प्रयोग	Punch Card	Batch OS	Mili Second	e.g., ENIAC, UNIVAC, EDVAC, IBM701
दूसरी पीढ़ी (1955-1969)	Transistor 	उच्च स्तरीय असेम्बली भाषा का प्रयोग जैसे- कोबोल, फ़ोरट्रान	Magnetic Tape and Punched Card	Time Sharing OS	Micro Second	e.g., CDC1604, IBM7030, UNIVAC 1107, HONEYMEN 400
तीसरी पीढ़ी (1969-1975)	IC (Integrated Circuit) 	उच्च स्तरीय भाषा का विकास जैसे- PL/1, Pascal, Basic	Large Magnetic Core	Remote OS	Nano Second	e.g., IBM360, 370, PDP8, CDC 6600

Generation	Hardware Components	Characteristics of Computer	Memory	Operating System	Speed	Example
4. चौथी पीढ़ी (1975-1989)	VLSI (Very Large Scale Integration) And Microprocessor 	C, SQL भाषा का प्रयोग तथा GUI आधारित O.S.का निर्माण	Magnetic Disk (HDD, FDD, DVD)	GUI Based Window, MAC	Pico Second	e.g., CRAY1, CRAY2, APPLE2, TRX80
5. पांचवी पीढ़ी (1989 से अब तक)	ULSI (Ultra Large Scale Integration Circuit) 	एआई (AI) का प्रयोग, इंटरनेट का प्रयोग, w.w.w का विकास	SSD (Solid State Drive)	Lisp OS	Femto, Hacto second	e.g., P.C. Tablet, Super Computer, IBM Notebook, Param 8000



\* नोट - विभिन्न स्रोतों में कम्प्यूटर जनरेशन की अलग-2 समयावधि बतायी गयी है। इनमें से सबसे एक्यूरेट(Accurate) डाटा का प्रयोग किया गया है।

# Computer Languages (कम्प्यूटर भाषाएँ)

कम्प्यूटर भाषा का उपयोग कम्प्यूटर सिस्टम में निर्देश देने के लिए किया जाता है। कम्प्यूटर मुख्यतः बाइनरी भाषा (0,1) समझता है। समझ के आधार पर कम्प्यूटर भाषा को तीन भागों में विभाजित किया जाता है जो निम्न हैं-

## 1-Low Level Language

ऐसी भाषा जो इंसानों को बिल्कुल भी नहीं जबकि कम्प्यूटर को पूर्णतः समझ आती है।

e.g.- मशीन लैंग्वेज, असेम्बली लैंग्वेज इत्यादि।

## 2-Middle Level Language

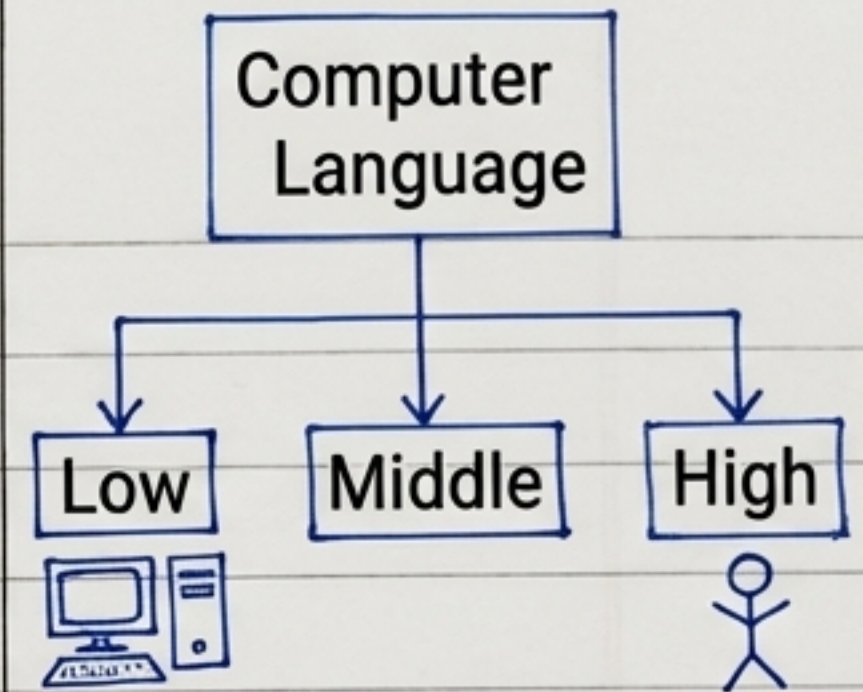
जो भाषा कम्प्यूटर के साथ-साथ इंसानों को भी समझ आ जाए।

e.g.- C, Pascal इत्यादि।

## 3-High Level Language

ऐसी भाषा जो इंसानों को पूर्णतः समझ आ जाए जबकि कम्प्यूटर को डायरेक्ट न समझ आए।

e.g.- C, C++, Python, Java, JavaScript, Ruby, C# etc.



प्रयोग के आधार पर कम्प्यूटर भाषा (प्रोग्रामिंग लैंग्वेज) को मुख्यतः दो भाग में विभाजित किया जाता है जो निम्न हैं-

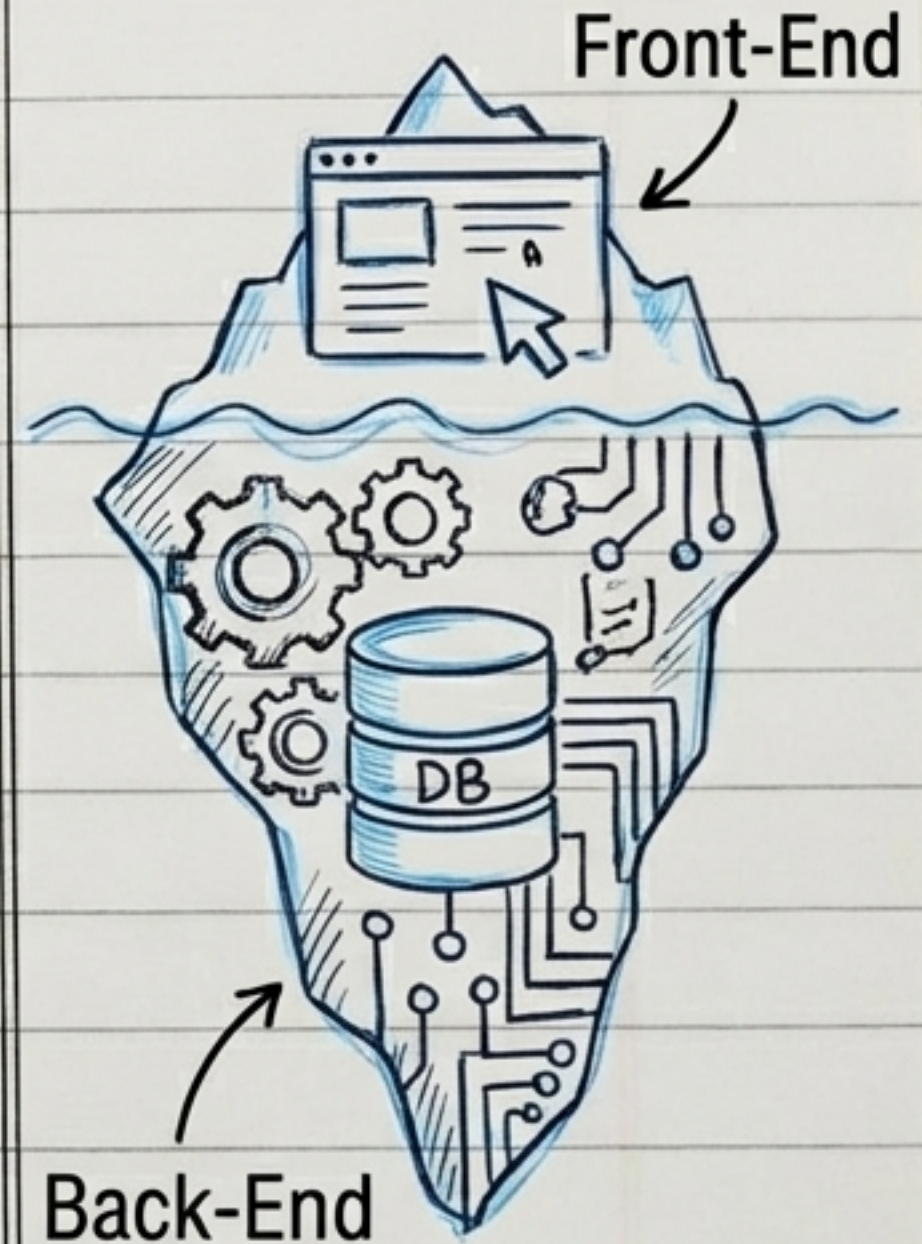
## 1- Front End

फ्रंट-एंड लैंग्वेज का उपयोग वेबसाइट या एप्लिकेशन का वह हिस्सा बनाने के लिए किया जाता है जो सीधे उपयोगकर्तों के सामने दिखता है और जिससे उपयोगकर्ता बातचीत (Interaction) कर सकता है।

e.g. HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), Bootstrap, JavaScript, jQuery, AngularJS, React, Swift etc.

## 2- Back End

कम्प्यूटर प्रोग्राम या किसी एप्लिकेशन के कोड के वे हिस्से जिनकी मदद से प्रोग्राम चलते हैं तथा वह यूजर को नहीं दिखते हैं।  
e.g. C, C++, Python, Java, PHP, C# etc.

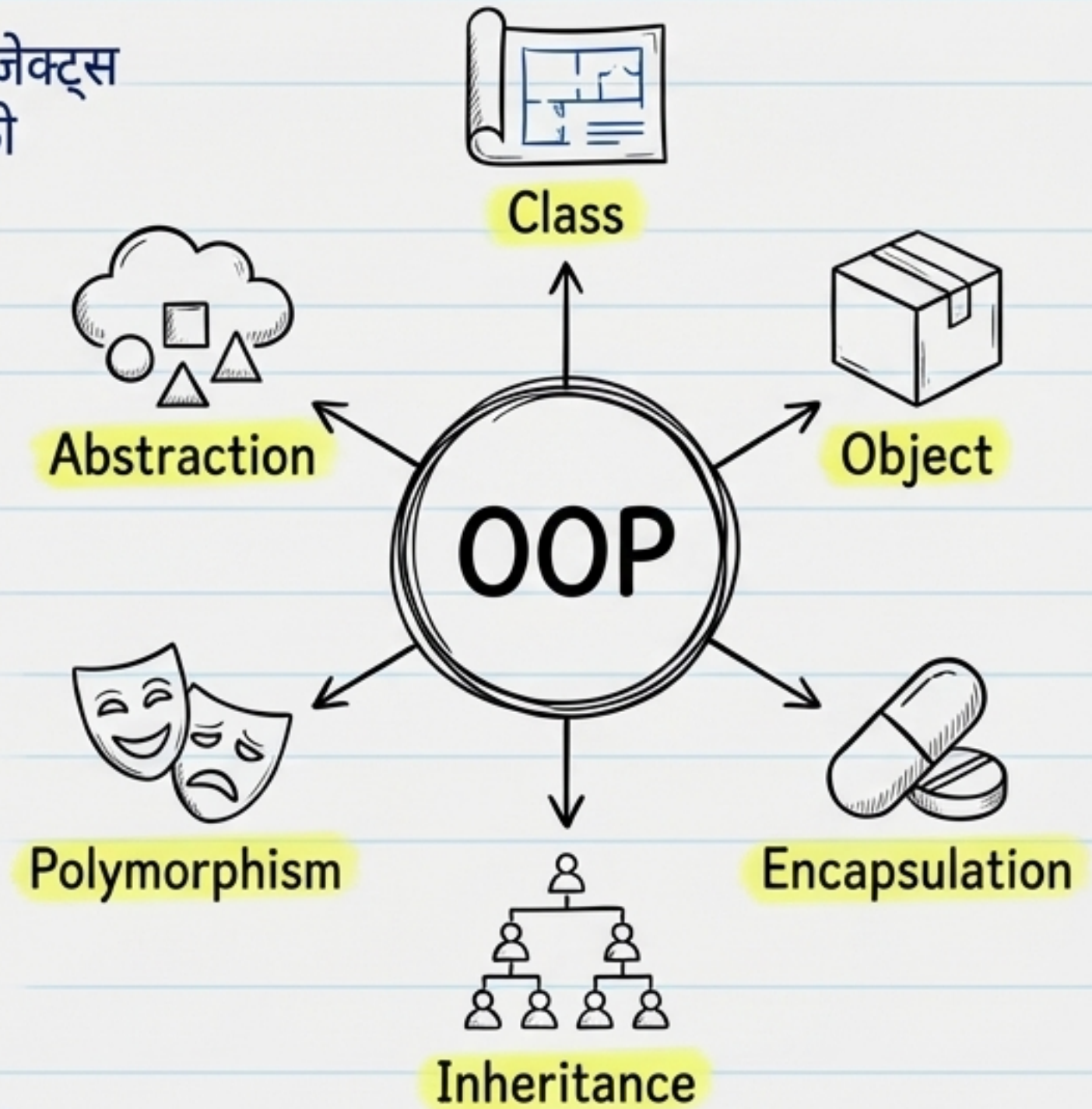


# Object-Oriented Programming (OOP)

Object-Oriented Programming (OOP) एक प्रोग्रामिंग शैली है जो ऑब्जेक्ट्स (Objects) और क्लासेस (Classes) के इर्द-गिर्द केंद्रित होती है। यह कोड को अधिक मॉड्यूलर, पुनः प्रयोज्य (Reusable), और स्केलेबल बनाता है।

## Features

- Class (क्लास)** - यह एक ब्लूप्रिंट (Blueprint) या टेम्पलेट होता है, जिससे ऑब्जेक्ट बनाए जाते हैं।
- Object (ऑब्जेक्ट)** - यह क्लास का एक वास्तविक उदाहरण (Instance) होता है।
- Encapsulation (एनकैप्सुलेशन)** - डेटा और उससे जुड़े मेथड्स को एक साथ क्लास के अंदर रखना एनकैप्सुलेशन कहलाता है।
- Inheritance (इनहेरिटेंस)** - एक क्लास दूसरी क्लास की प्रॉपर्टीज और मेथड्स को प्राप्त कर सकती है।
- Polymorphism (पॉलीमॉर्फिज्म)** - एक ही मेथड अलग-अलग कार्य कर सकता है।
- Abstraction (एब्स्ट्रैक्शन)** - अनावश्यक विवरण छुपाकर केवल महत्वपूर्ण जानकारी प्रदान करता है।



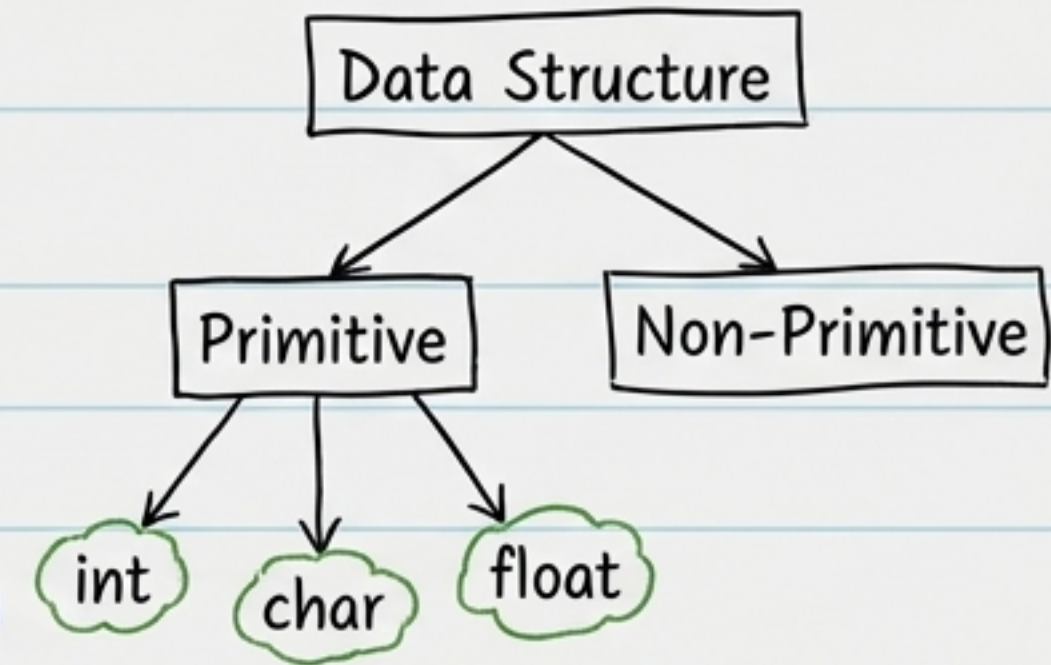
# Data Structure

Data structure (डेटा स्ट्रक्चर) एक विशेष तरीके से डेटा को संगठित (organize) और स्टोर करने का तरीका है, ताकि उसे प्रभावी ढंग से एक्सेस (access) और संशोधित (modify) किया जा सके। यह डेटा के बीच संबंधों (relationships) को परिभाषित करता है। यह दो प्रकार के होते हैं - i. Primitive ii. Non Primitive

## (i). PRIMITIVE

प्रिमिटिव (primitive) का मतलब मूलभूत या प्रारंभिक होता है। कंप्यूटर विज्ञान में, प्रिमिटिव डेटा प्रकार (Primitive data Type) वे बुनियादी डेटा प्रकार होते हैं जो सीधे सिस्टम द्वारा सपोर्ट किए जाते हैं और अधिक जटिल डेटा संरचनाओं का निर्माण करने के लिए उपयोग किए जाते हैं।

*e.g.- int, char, float, Boolean etc*



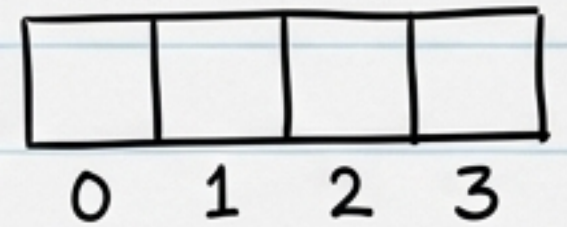
## (ii). NON PRIMITIVE

नॉन-प्रिमिटिव डेटा प्रकार वे डेटा प्रकार होते हैं जो प्रिमिटिव डेटा प्रकारों से मिलकर बनते हैं। ये अधिक जटिल होते हैं और इनके द्वारा बड़ी मात्रा में डेटा को संगठित और संग्रहित किया जा सकता है।

e.g.- array, object, string, structure, union, linked list, stack, queue etc

### → Array

समान प्रकार (Homogenous) व निश्चित आकार के डेटा तत्व को सारणी के रूप में क्रमबद्ध (linear) संग्रह करता है। इसमें डेटा तत्व को एक्सेस करने के लिए index का प्रयोग किया जाता है जो 0 से शुरू होता है।



### → Stack

यह LIFO (Last In First Out) के सिद्धांत पर कार्य करता है। इसका मतलब जो आइटम सबसे अंत में जोड़ा गया है वही सबसे पहले हटाया जाता है।

e.g प्लेट को एक छल्ली में रखना।



इसमें मुख्यतः तीन क्रियाएँ होती हैं -

i- *Push* - नया तत्व जोड़ना (टॉप पर)

ii- *Pop* - शीर्ष तत्व को हटाना

iii- *Peek* - टॉप *element* को देखना

### → Queue (क्यू)

यह *FIFO (First In First Out)* के सिद्धांत पर काम करने वाली डाटा संरचना है इसका मतलब जो तत्व सबसे पहले जोड़ा गया है वही हटाया जायेगा।

e.g - भण्डारे व रेलवे स्टेशन की लाइन etc

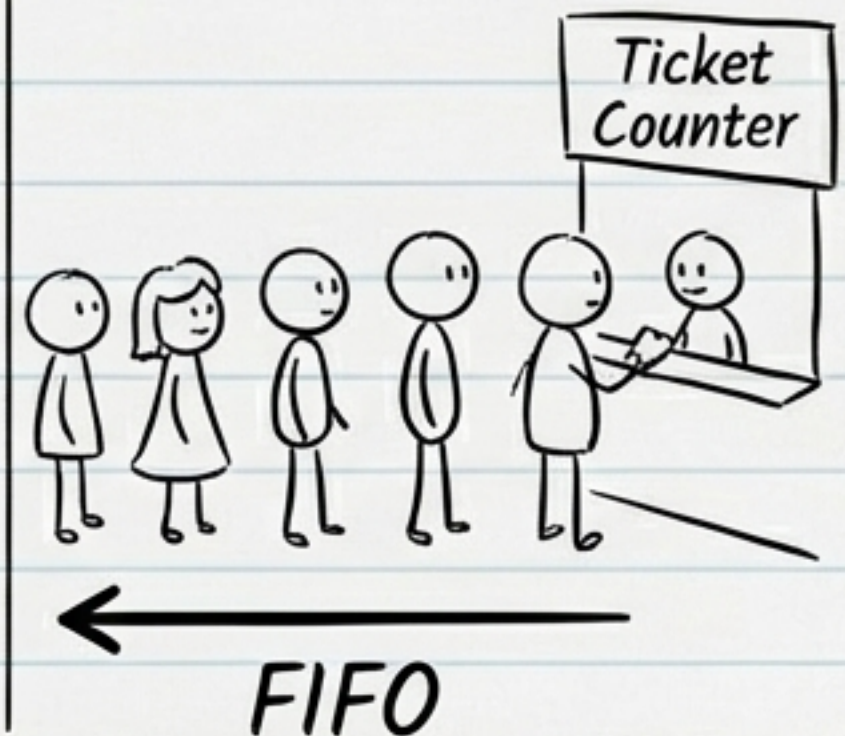
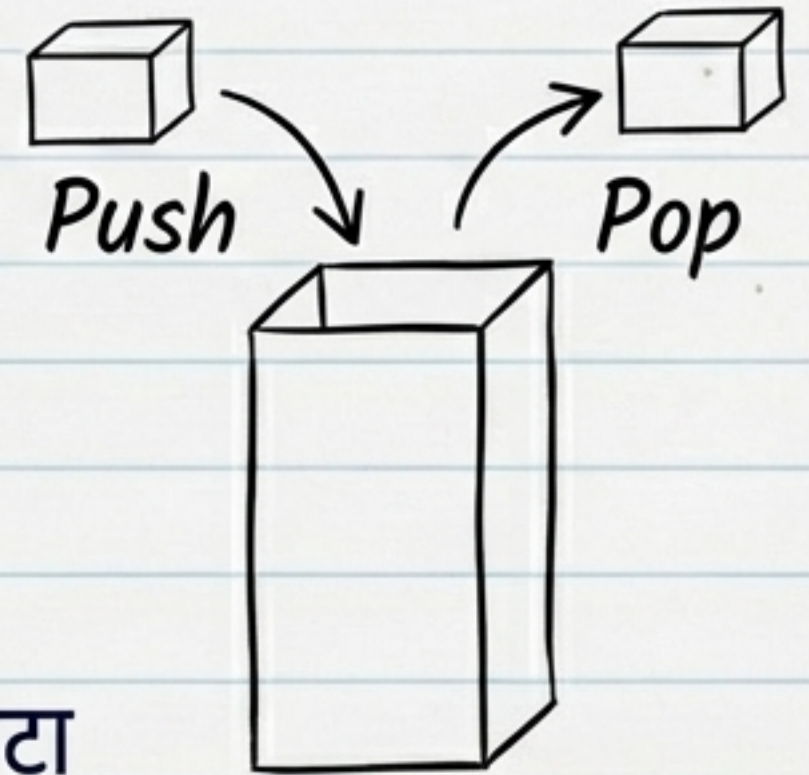
इसमें मुख्यतः चार क्रियाएँ होती हैं -

i- *Enqueue* - नया तत्व जोड़ना

ii- *Dequeue* - सबसे पहले जोड़ा गया तत्व हटाना

iii- *Front* - सबसे आगे के तत्व को देखना

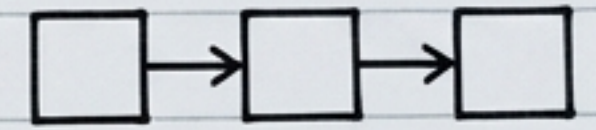
iv- *Rear* - सबसे पीछे वाले तत्व को देखना



→ **Linked List**

यह एक डायनेमिक डेटा संरचना है जिसमें नोड का प्रयोग कर के डेटा संग्रहित किया जाता है।

e.g - ट्रेन के डिब्बे etc

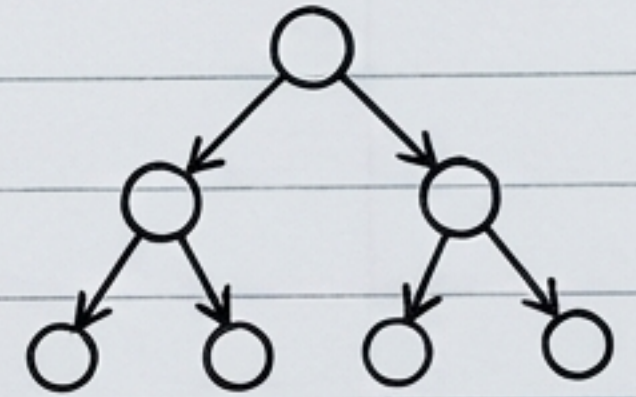


Linked List

→ **Tree (ट्री)**

यह Non Linear पदानुक्रमित (Hierarchical) डेटा संरचना है। जिसमें डाटा nodes में store होता है।

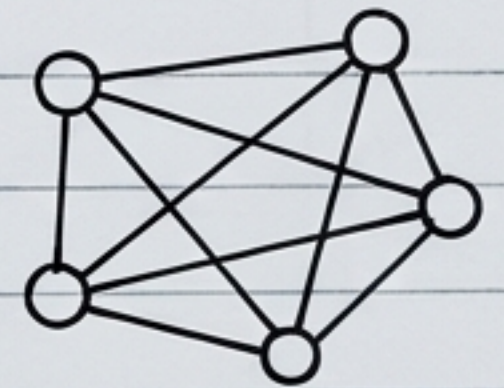
eg. - बाइनरी ट्री, AVL ट्री etc.



Tree

→ **ग्राफ (Graph)**

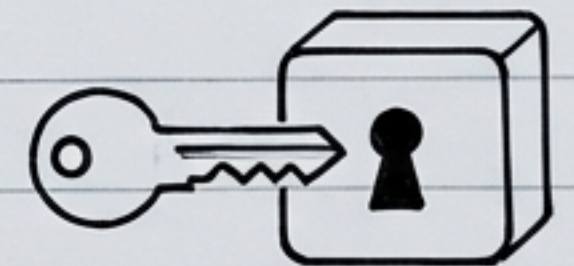
यह एक Non Linear संरचना है जिसमें नोड्स और किनारों का उपयोग कर के डाटा को स्टोर करते हैं।



Graph

→ **हैश टेबल**

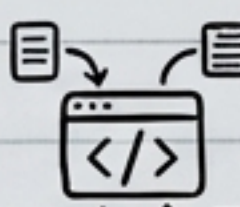
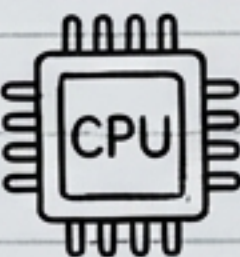
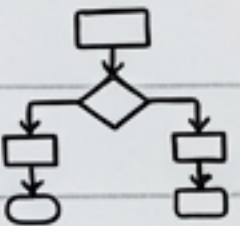
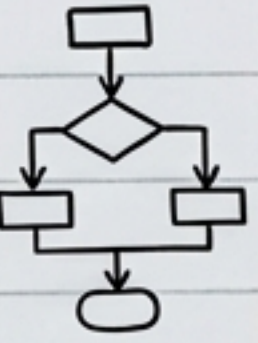
यह डेटा को एक key के आधार पर संग्रहित करता है तथा हैशिंग तकनीक का प्रयोग करके डेटा को संगठित करता है। यह एक Non Linear संरचना है।



Hash Table

# IMPORTANT COMPUTER LANGUAGES

कम्प्यूटर भाषा	वर्ष	डेवलपर/संस्थापक	मुख्य उपयोग
मशीन भाषा (Machine Language)	1940	-----	यह सबसे निम्न स्तर की भाषा (Low Level Language) है जो सीधे हार्डवेयर के साथ काम करती है। इसे बाइनरी कोड (0s और 1s) के रूप में लिखा जाता है और यह तेज़ होती है, लेकिन इसे समझना और उपयोग करना कठिन है।
असेम्बली भाषा (Assembly Language)	1949	कैथलीन बूथ (Kathleen Booth)	यह मशीन भाषा का प्रतीकात्मक रूप है, जिसमें मानव-पठनीय कोड का उपयोग होता है। यह हार्डवेयर संचालन को नियंत्रित करने के लिए उपयोग की जाती है और मशीन-स्तरीय प्रोग्रामिंग में महत्वपूर्ण होती है।
फॉर्ट्रान (FORTRAN)	1957	जॉन बेकस (John Backus)	यह वैज्ञानिक और गणितीय गणनाओं के लिए एक उच्च स्तरीय भाषा है। फॉर्ट्रान का उपयोग इंजीनियरिंग, गणित और भौतिकी के अनुप्रयोगों में होता है। यह पुरानी लेकिन शक्तिशाली भाषा है।
कोबोल (COBOL)	1959	ग्रेस हूपर (Grace Hopper)	यह व्यावसायिक डेटा प्रोसेसिंग के लिए डिज़ाइन की गई है। कोबोल का उपयोग बड़े उद्यमों, बैंकों और सरकारी संस्थानों के डेटाबेस प्रबंधन और ट्रांजेक्शन सिस्टम में किया जाता है। यह प्रथम कम्प्यूटर भाषा है।



# IMPORTANT COMPUTER LANGUAGES

कम्प्यूटर भाषा	वर्ष	डेवलपर/संस्थापक	मुख्य उपयोग
सी (C)	1972	डेनिस रिची (Dennis Ritchie)	यह प्रणालीगत सॉफ्टवेयर और ऑपरेटिंग सिस्टम्स के विकास के लिए उपयोग की जाने वाली एक सामान्य उद्देश्य वाली भाषा है। इसे पोर्टेबल, तेज़ और कार्यक्षमता-समृद्ध होने के कारण महत्वपूर्ण माना जाता है।
सी++ (C++)	1983	ब्यार्ना स्ट्रोस्ट्रुप (Bjarne Stroustrup)	यह ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग के लिए विकसित की गई है। C++ का उपयोग गेमिंग, ग्राफिक्स और सिस्टम प्रोग्रामिंग में किया जाता है। यह C की कार्यक्षमता को विस्तारित करता है।
जावा (Java)	1995	जेम्स गोसलिंग (James Gosling)	जावा एक ऑब्जेक्ट-ओरिएंटेड, प्लेटफार्म-इंडिपेंडेंट भाषा है जिसका उपयोग वेब और मोबाइल एप्लिकेशन के विकास में किया जाता है। इसे इसकी "Write one and Run Anywhere" की क्षमताओं के लिए जाना जाता है।
पायथन (Python)	1991	गुइडो वैन रोस्सम (Guido van Rossum)	यह एक उच्च स्तरीय, इंटरप्रेटेड और कम्पाइल्ड (आंतरिक) भाषा है जिसका उपयोग डेटा साइंस, मशीन लर्निंग, वेब डेवलपमेंट और स्वचालन में किया जाता है। पायथन अपनी सरलता और पढ़ने में आसान सिंटैक्स के लिए लोकप्रिय है।

End of Notes

