First Generation Languages.

Example: 01.

Machine language for text "Hello world"

01001000 01100101 01101100 01101111

00100000 01010111 01110010 01101100

01100100

Example: 02

2 + 5 = 7

10 + 101 = 111

Second Generation Languages.

Assembly language.

Example 01:

"Hello world" written for 32-bit intel processor.

Scanned with CamScanner

```
global    -main
extern    -printf
section   .text

-main:

Push      message
call      -printf
add       esp, 4

message:

db 'Hello, World!', 10,0
```

Types of Assembly languages

CISC
RISC
DSP
VLIW

- Each Assembly instruction is translated into machine language instruction.

Third Generation Languages.

FORTRAN – IBM 1950s, scientific and Engineering apps.

```
Program hello
    print *, "Hello world!"
end Program
```

Fortran is case sensitive, give the

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLOWRD.
PROCEDURE DIVISION.
DISCPLAY" HELLO WORLD".
STOP RUN.
```

business language used by corporate, Govt sectors.

— x —

C. – Dennis Ritchie in 1972 at bell lab

```
#include<stdio.h>
int main() {
// printf display the string inside
the quotation.
Printf("Hello World");
return 0;
}
```

C++ – by Bjarne Stroustrup in 1979 at bell lab.

```
#include<iostream>
int main() {
std::cout<<" Hello World";
return 0;
}
```

Scanned with CamScanner

Visual Basic. by Microsoft in 1991

```
Imports System
Module Module1 ——> this is module name.
    Sub Main()
        Console.WriteLine("Hello World")
        Console.WriteLine("Press Enter to exit")

        Console.ReadLine()

    End Sub

End Module
```

——— x ——— x ——— x ———

Fourth Generation Languages.

Example :

```
SELECT * FROM Customers;
```

The above mentioned statement selects all the columns from the "Customers" table.

Scanned with CamScanner

Programming Paradigms.

1. Sequential

- Execution and results are same as the sequence of instruction.

Characteristics:

- Order of execution is determined by the textual order of the statement.
- Sequence of the written statements will be executed without any overlap.

Example 01:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b,c;
    float avg;
Printf("Enter Numbers:\n");
scanf("%d %d %d", &a, &b, &c);
avg = a+b+c/3;
printf("Average", avg);
    getch();
}
```

header file

Function that doesn't return any value.

it contains functions prototype like printf(), scanf() so that compiler knows what are their parameters and return values.

Scanned with CamScanner

Procedural Paradigm.

```
#include <math.h>
# include < stdio.h >
int main( ) {
double number , squareroot;
Printf(" Enter number");
scanf( " %lf", & number);
// computing the square root
squareroot = sqrt(number);
Prinf(" Square Root is :" , number,
                squareroot);

return 0;
}
```

→ used by main function and main function should return 0 if program has executed successfully, otherwise -1.

Scanned with CamScanner

Lec # 03

Intro to object oriented pardigm
Principles of object oriented paradigm

What is object orientation?

A technique for system modeling
object oriented model consist of several interacting objects.

What is a model?

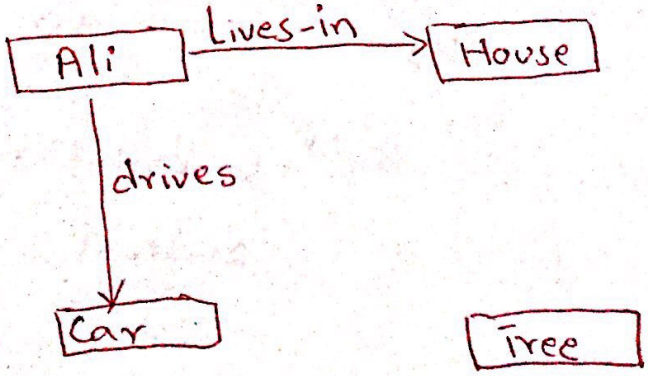
abstraction of something,
Something real, something conceptual.

Why do we need a model?

To understand the product before developing it. (e.g Architectural, maps, Mechanical),

01 - Example - OO Model

An imag of the model is mentioned at slide no. 21 of week1.pdf lecture uploaded at google classroom



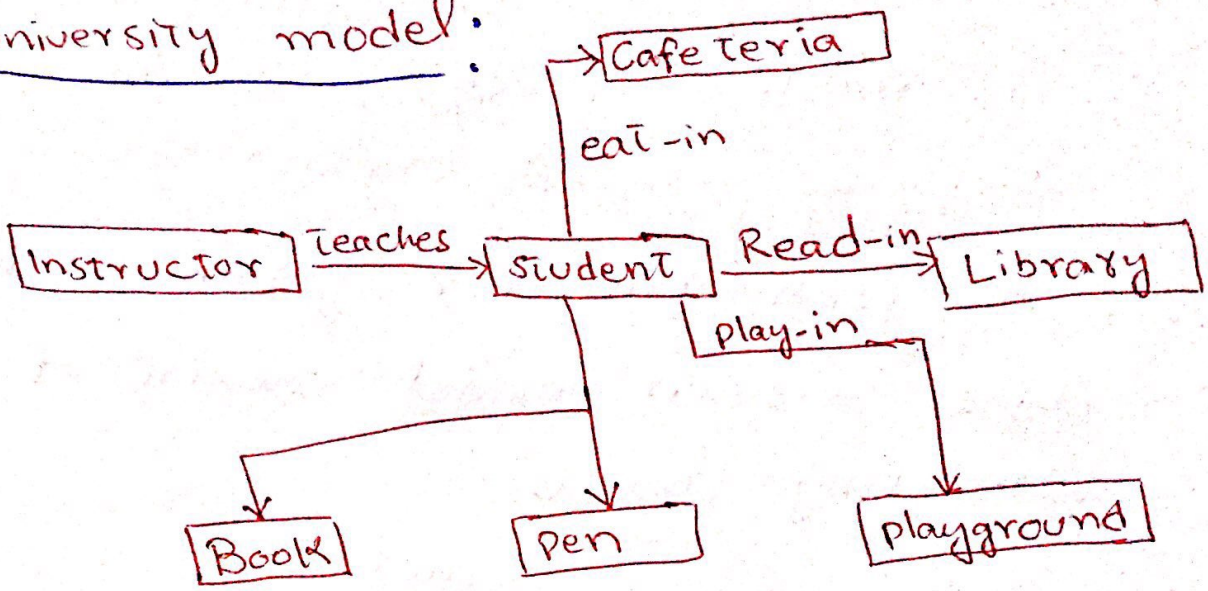
objects
person
House
Car
Tree

Interactions
Ali lives in the house
Ali drives the car.

Scanned with CamScanner

Objects in a school/college/Uni

A university model:



Objects:	behavior/Interactions:
Instructor	Instructor teaches to students
Student	Student read in Library
Library	Student play in playground
Cafeteria	Student eat in cafeteria
Book	Student has pen and book.
Pen	
Playground	

Object has an identity

for e.g

Instructor has a name/id.

Student has a name/id.

Scanned with CamScanner

Advantages of object Orientation

- object oriented model map to reality
- We think in terms of objects.
- Easy to implement an oo model for a given Problem.

Object:

Something Tangible (Ali, car, House, student, instructor, pen, Book)

Something conceptual (Date, Time and so on)

An object has state (attributes) /(data)/ Properties like (Name, id, Age, height)

An object has well defined behavior/methods/ functions (walks, playing, reading, eating)

Conceptual object Example

Time

state (attributes)	operations | behavior
Hours	set hour
second	set seconds
minutes	set minutes.

Scanned with CamScanner

Principles of object oriented paradigm.

Abstraction
Encapsulation
Polymorphism
Inheritance
Reusability

} Discussion in class

Car as an object.

As class activity by students.

Scanned with CamScanner