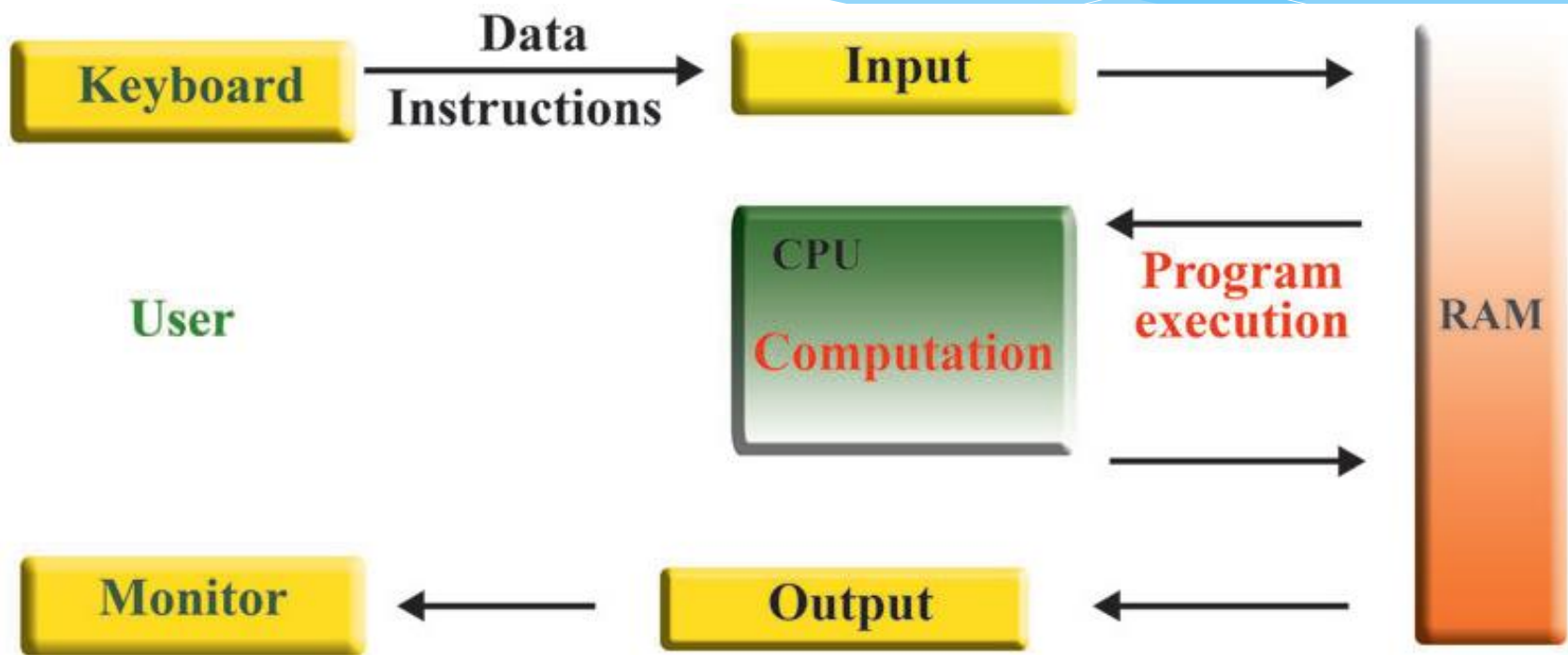


Evolution of programming languages

Need of a programming language

program

- * A sequence of instructions
- * designed to perform a certain task



Low level programming languages

- * **Machine language**

- * **Assembly language**

features of machine language program ;

- * Could be executed directly on the machine
- * Fast in operation
- * No need of language translating programs to translate the program into binary
- * Dependency on machines (a program written to one computer may not run on another computer)
- * Difficult to understand by humans as it is written

features of assembly language program ;

- * Operation is comparatively slower than the machine language.
- * Assembly language should be translated to instructions using the language translating program called assembler.
- * Dependency on machines (a program written for one computer cannot be run on another computer.)
- * The use of symbols makes it more simple to understand.

**A program written in
Assembly language**



Assembler



**Machine language
instructions**

High-level programming languages

Languages which are designed with simple English words enabling the programmer to understand it easily

Eg:- **FORTRAN, BASIC, COBOL, PASCAL, C, JAVA, VB, PYTHON**

features of high-level language

- * Easy to understand.
- * Need to be translated to machine language instructions before executing on a computer.
- * Does not depend on the machine. .

Language translators

A program which translate computer programs/instructions into machine language

Two types of language translators

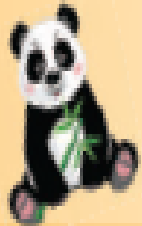
1. Interpreter
2. Compiler

Interpreter

translates each statement written in a high-level computer language to machine language commands one by one and the translated program is executed using the necessary commands instantly.

1. If there are no syntax errors in the program, the statement will be executed
2. If there are syntax errors in the program, it will not execute to the end. (it is possible to operate it till the error is reached)

Important



An interpreter translates code each time the program is executed.

A program written in
high level language



Interpreter



Machine language
instructions

Compiler

translates the entire program written in a high level language to machine language as a whole, before it could be executed.

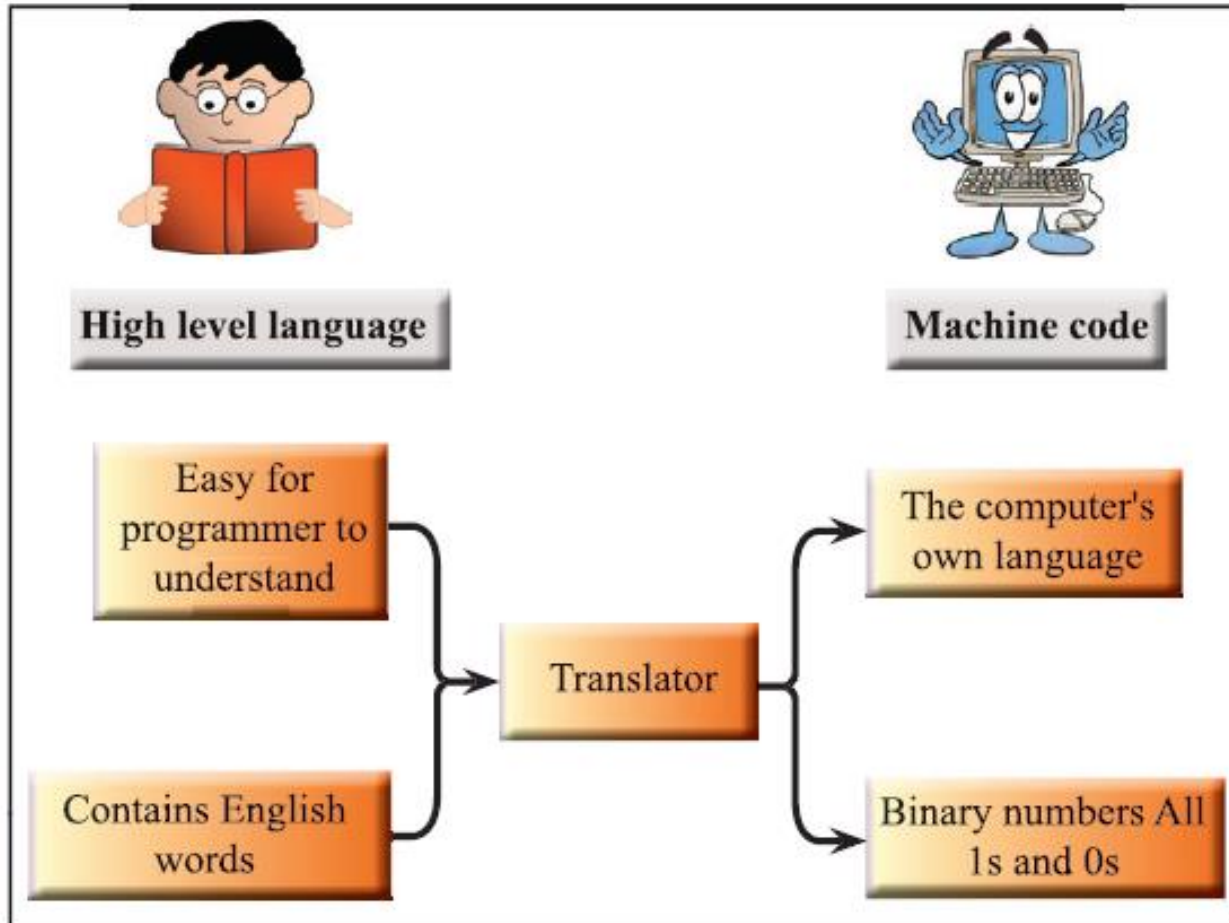
- * If there are no syntax errors in the program, code is translated to machine code.
- * If there are syntax errors in the program, it is not be translated to machine code. These errors are highlighted.

Important



After the program is translated to machine code once, it can be execute any number of times. A translation is needed again only if the source code is changed.





programming paradigms

Different approaches to computer programming are called programming paradigms.

- * Different approaches develop solutions to problems using programs using different paradigms.

Paradigms

Imperative

Declarative

Procedural

**Object
Oriented**

**Parallel
Processing**

Logic

**Functional
Dataflow**

Database

• C

• Java

• Java

• Prolog

• Lisp

• SQL

Programming Paradigms

Imperative/ Algorithmic	Declarative		Object- Oriented
	Functional Programming	Logic Programming	
Algol Cobol PL/1 Ada C Modula - 3	Lisp Haskell ML Miranda APL	Prolog	Smalltalk Simula C++ Java

Difference between procedural and declarative paradigms

- * A **procedural language** is a computer programming language which consists of a well structured set of steps and procedures. This includes statements for problem solving steps. (Eg: Pascal)
- * A **declarative paradigm** develops a structure and elements of computer program by indicating calculations and/or logic without a control flow. program is designed to solve problems explaining what you want rather than stating how to solve the problem as in primary programming languages. Declarative paradigms related to Artificial Intelligence.

Important



Procedural Paradigm

Saying how you achieve it



1. Keep block A
2. Keep block B on top of block A
3. Keep block C on top of block B

Declarative Paradigm

Saying what you want



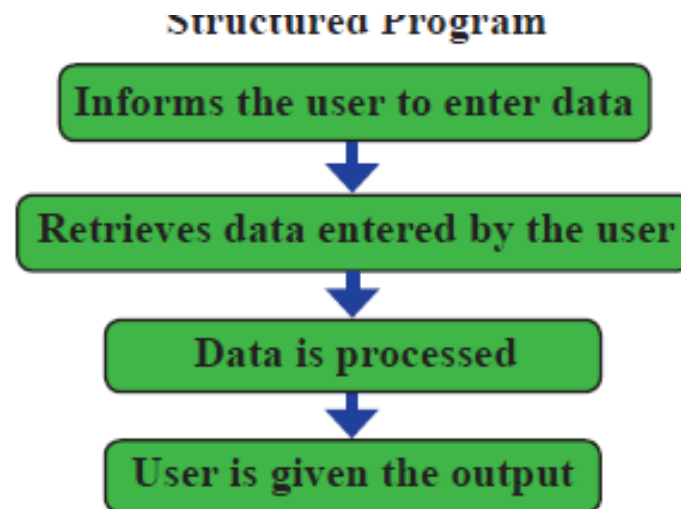
A pillar which consists of 3 blocks



Comparison of structured and object oriented paradigms

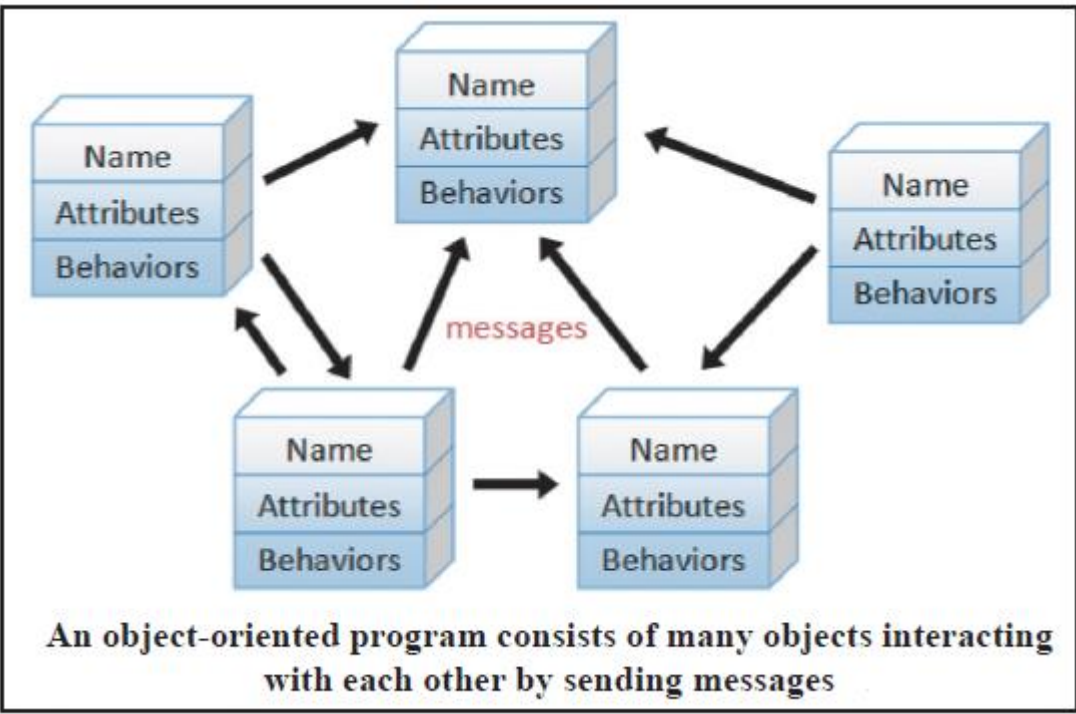
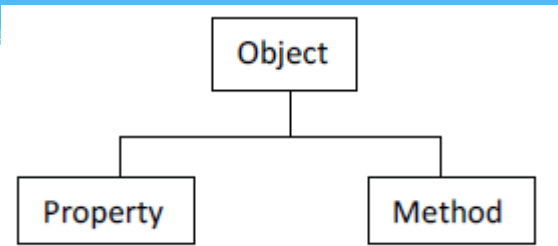
Structured paradigm

- * A structured program is a logic based paradigm and it is the pre discussed object oriented program.
- * provides facilities to understand and modify the program.
- * system is divided into sub systems and there is a top – down flow in it



Object oriented programming

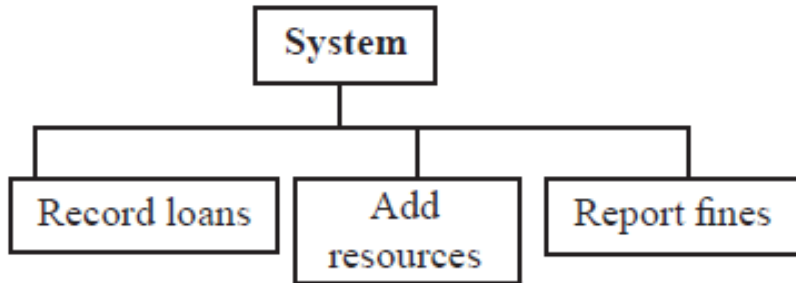
- * based on the concept of **objects**. Objects consists of data and methods. Methods are codes that are in the form of procedures that handle data.
- * These data structures exist in the form of fields which are called **Attributes**.
- * Class is the basic structure of object oriented programming.
- * Class describes the behaviors of data and instances.
- * **Class** can create objects of same type.



Library Information Systems

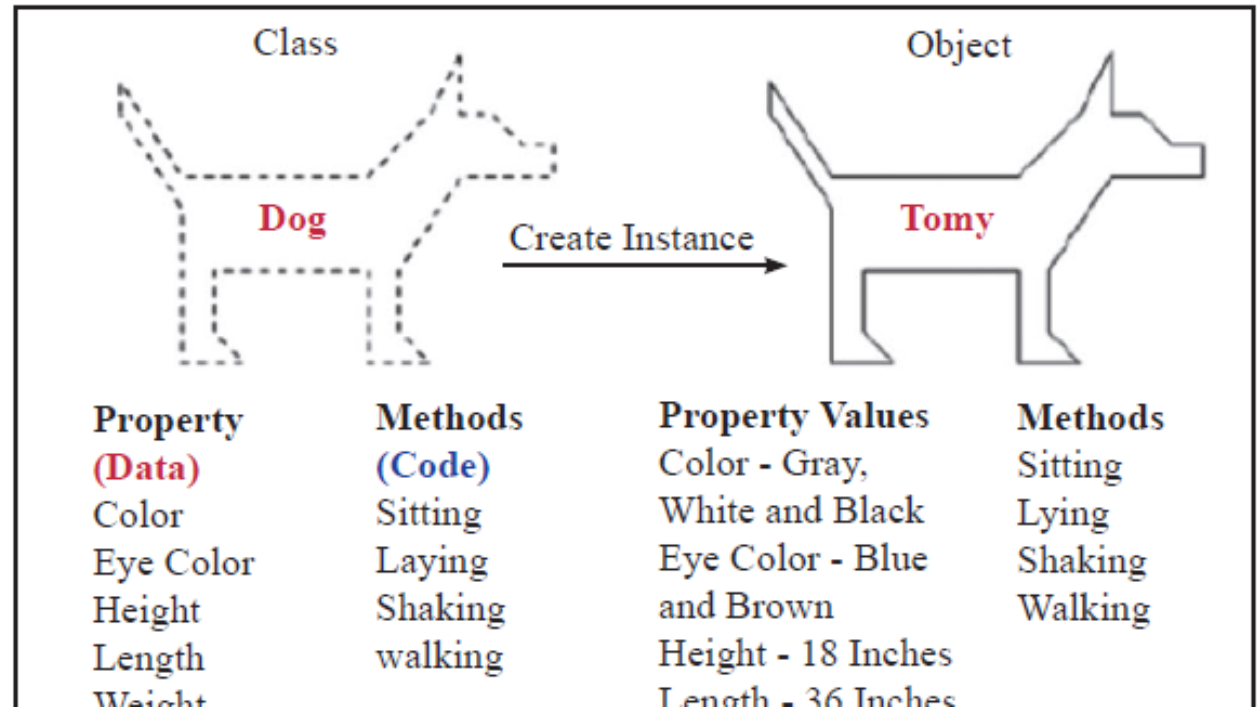
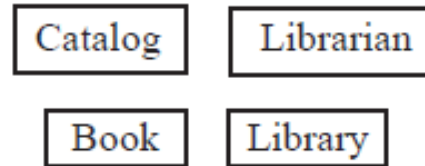
Structured Approach

Decompose by functions or processes



Object Approach

Decompose by objects or concepts



Programming and scripting

Programming languages	scripting languages
hard to code	Easier to code
fast	slow
Compiler based language	Interpreter based language
Run independently	Run inside another program
JAVA,VB	JavaScript and PHP are scripting