28/8/23
Monday

Page No. 1
Date

UNIT 01 INTRODUCTION TO PROGRAMMING PARADIGM

- Why learn programming language ?

→ To improve your ability to develop effect algorithms
→ To improve your use of your existing programming language.
→ To increase your vocabulary of useful programming constructs.
→ To allow a better choice of programming language.
→ To make it easier to learn a new language.
→ To make it easier to design a new language.

- Problem Solving requires following steps because the qualities of the language for expressing the solution must be such that the persons spends more time in solving problem and providing optimal solution :

1) Define the system
2) Analyse the defined system
3) Detail system specification
4) Design the system
5) Implement the design
6) Testing and debugging
7) Validation

- Types of Programming Paradigms

1) Structured Programming / Structural (C)
2) Object Oriented Programming (Java, C++)
3) Functional Programming (LISP)

4) Logic Programming (PROLOG)
5) Event driven Programming
6) Concurrent Programming
7) Distributed Programming
8) Database Programming

30/8/2023
Tuesday

- If programming paradigm isn't followed :-
  1) Increased complexity
  2) Less readability
  3) Improper structure
  4) Hard to test
  5) Hard to change
  6) Maintaining will be hard

- Programming Paradigm

Imperative	Declarative
1) Procedural and object oriented	1) functional & logical programming paradigm

- Structured Programming / Procedural Programming

→ Key Features of C Programming :-
1) Sequential ~~exquistior~~ execution of instructions
2) Use of variables representing memory locations
3) Use of assignment of change values of variables
4) Conditional branch & iterative statements
5) Recursion as an alternative to iteration.

→ Functions have return type but procedures don't.
* Difference between function & procedure ? *

→ goto statements are used in structured programming
→ Procedures aren't functions as functions return values and procedures don't.
→ C, Pascal

• Object Oriented Programming
→ Java, C++, Smalltalk, SNOBOL
→ Classes & Objects :-
Objects are basic building block and are equal to real life entities.
An object is characterised by state and behaviour. The state is specified by attributes and behaviour is specified by methods.
Encapsulation, polymorphism, inheritance are the foundational concepts that give an identity to the paradigm.

→
```
public class calculator
{
  static int i=10;
  static int j=10;
  public static void main (string args[])
  {
   Addition a=new Addition();
   Subtraction s=new Subtraction();
   system.out print(a.add(i,j));
   system.out print(s.sub(i,j));
  }
}
class Addition
{
```

```
public int add (int i, int j)
{
  return i+j;
}
}
class Subtraction
{
 public int sub (int i; int j)
 {
   return i-j;
 }
}
```

* Declarative Programming
→ We tell the compiler what to do rather than how to do.
→ Functional Programming & Logic Programming are part of it.

07/09/23
Monday

→ In functional programming
Eg. LISP
The basic building block is function.
Recursion is a key facility.
→ In logic programming
Eg. PROLOG
It is based on symbolic logic.
A logic program is a collection of declarations that are true about desired result these are call facts

A set of rules that operate on facts are defined a query reports the results brought from the facts and covered by rule base.
or government
The inference engine ensures the validity of the result.

→ PROLOG Eg. :-

man (Socrates)	Socrates is a man
mortal (X) :- man (X)	All men are mortal
?- mortal (socrates)	Is socrates mortal ?
	O/P :- Yes

Language standardization

→ 1) Proprietary
2) Consensus

→ Proprietary standards :-
These are definitions by the company that develop and owns the language.
They don't work for languages that have become popular and widely used. Variations in implemen^n soon occur with many enhancements & incompabilities.

→ Consensus standards :-
These are documents produced by an organisation based on an agreement by the relevant participants.
They are major method to ensure uniformity among several users of the language.

→ Why standardization ?
Avoid complexity for programmers

→ Standardization means set of protocols to be followed while programming.

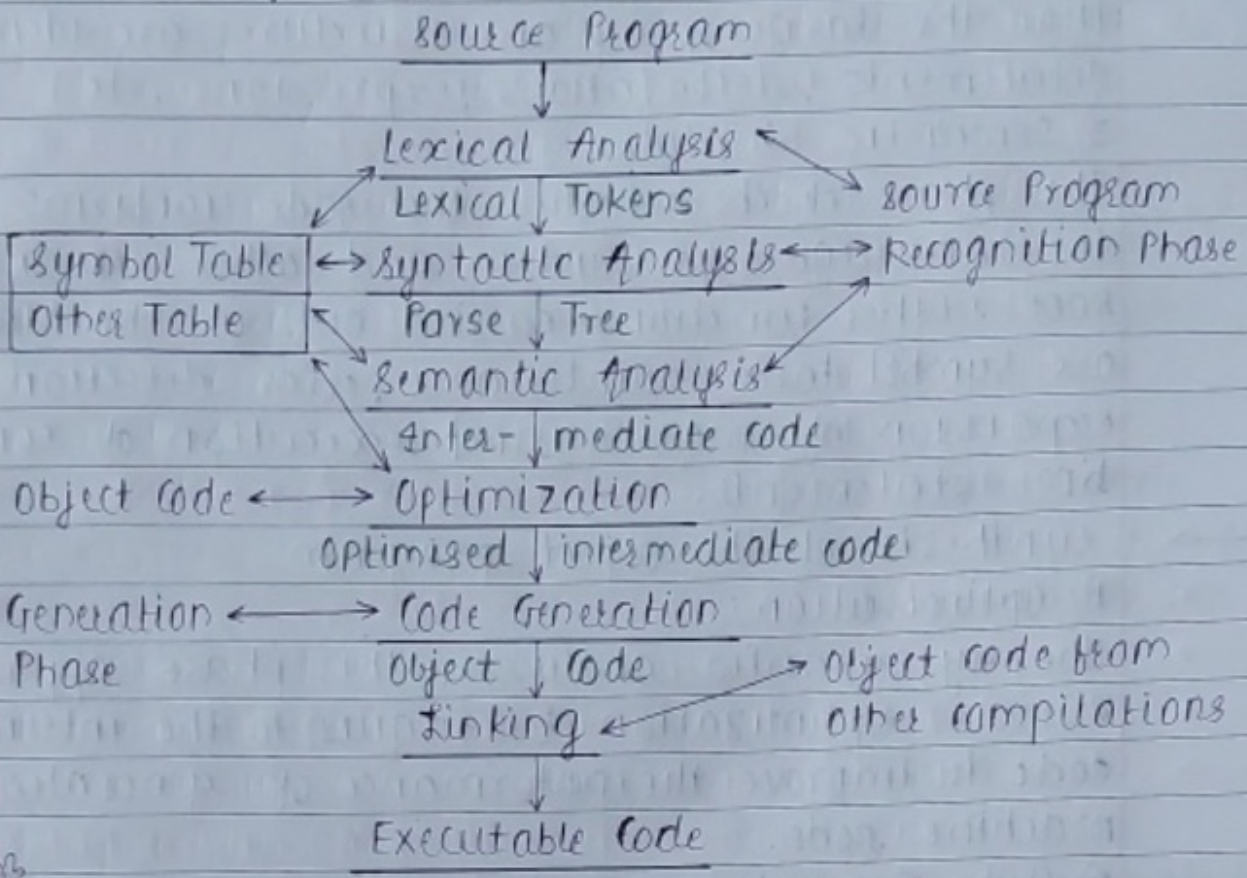
05/9/23
Tuesday

Page No. 6
Date

→ Things to remember for standardization
1) Timeliness
2) Conformance
3) Obsalescene
To use standards effectively three issues need to be addressed which are mentioned above.
Timeliness :-
It deals with when to standardise a language. Language should be standardised early enough so that there is enough experience in using the language but not so late to encourage many incompatible implementations.
Conformance :-
A program is conformant if it only uses features defined in the standard. A conforming compiler is one which when given a conformant program produces an executable program that produces correct output.
Obsalescene :-
The standardization process have to be reviewed every five years. They can be either be renewed or droped. Fortren language was standardized in 1966, revised in 1978 and renewed in 1990.

Stages in Translation
→ 1) Analysis
2) Synthesis
→ Code Optimization
Program tokens
Processor

→ Structure of Compiler :-



11/09/2023
Monday

→ Analysis of ~~of~~ the source program

1) Lexical Analysis

It is also called as scanning. Here the translatn phase groups sequence of characters into elementary constituents: identifiers, delimitors, operator symbol, number, keywords, noise keywords, blanks, comments etc. The lexical analysiser is the input routine for the translator reading successive lines of input program, breaking it down into individuals atoms called lexens or tokens and feeding these lexenes to the later stages.

2) Syntactic Analysis (Parsing)
Here the larges program structures are identified (statements, declarations, expressions, etc.)
3) Semantic Analysis
This phase checks whether the code confirms to the language type system and other semantic roots. Other functions carried out in the stage are symbol table maintenance, error detection, expansion of macros and execution of compile time statements.

→ Synthesis of the object program
1) Optimisation
The ~~sy~~ semantic analyses. This phase applies various optimization techniques to the intermediate code to improve the performance of generated machine code.
2) Code Generation
This phase takes optimised intermediate code and generates actual machine code that can be executed by target
3) Linking & Loading
The linker or loader loads the various segments of translated code into memory and then uses the attached loader table filling the data & sub program addresses the code as condition. The result is final executable program ready to run.

12/09/23
Tuesday

Page No. 9
Date

UNIT 02 INTRODUCTION TO JAVA PROGRAMMING

- History
→ Java was priorly called as Oak.
→ James Ghosling, Patrick Naughton, Mike Sheradin, Bill Joy :- invented Java
→ Owned by Sun Microsystems
→ Scientists who invented Java drank coffee made from coffee beans brought from Java island. That's why they renamed 'Oak' to 'Java' & java file has icon of coffee mug.

C++	Java
1) Not purely OOP	1) Purely OOP
2) Pointers don't exist	2) Pointers are eliminated because they can compromise system security because they store memory locations
3) goto() exists	3) goto() doesn't exist
4) Multiple Inheritance ✓	4) Multiple Inheritance X
5) Access specifiers (3) :- Public, Private, Protected	5) Access specifiers (4) :- Public, Private, Protected, Default
6) Allocation & deallocatn of memory is done by programmer	6) Allocation & deallocation of memory is done by JVM (Java Virtual Machine)
7) Operator Overloading ✓	7) operator Overloading X
8) Constructor ✓	8) Constructor ✓
Destructor ✓	Destructor X

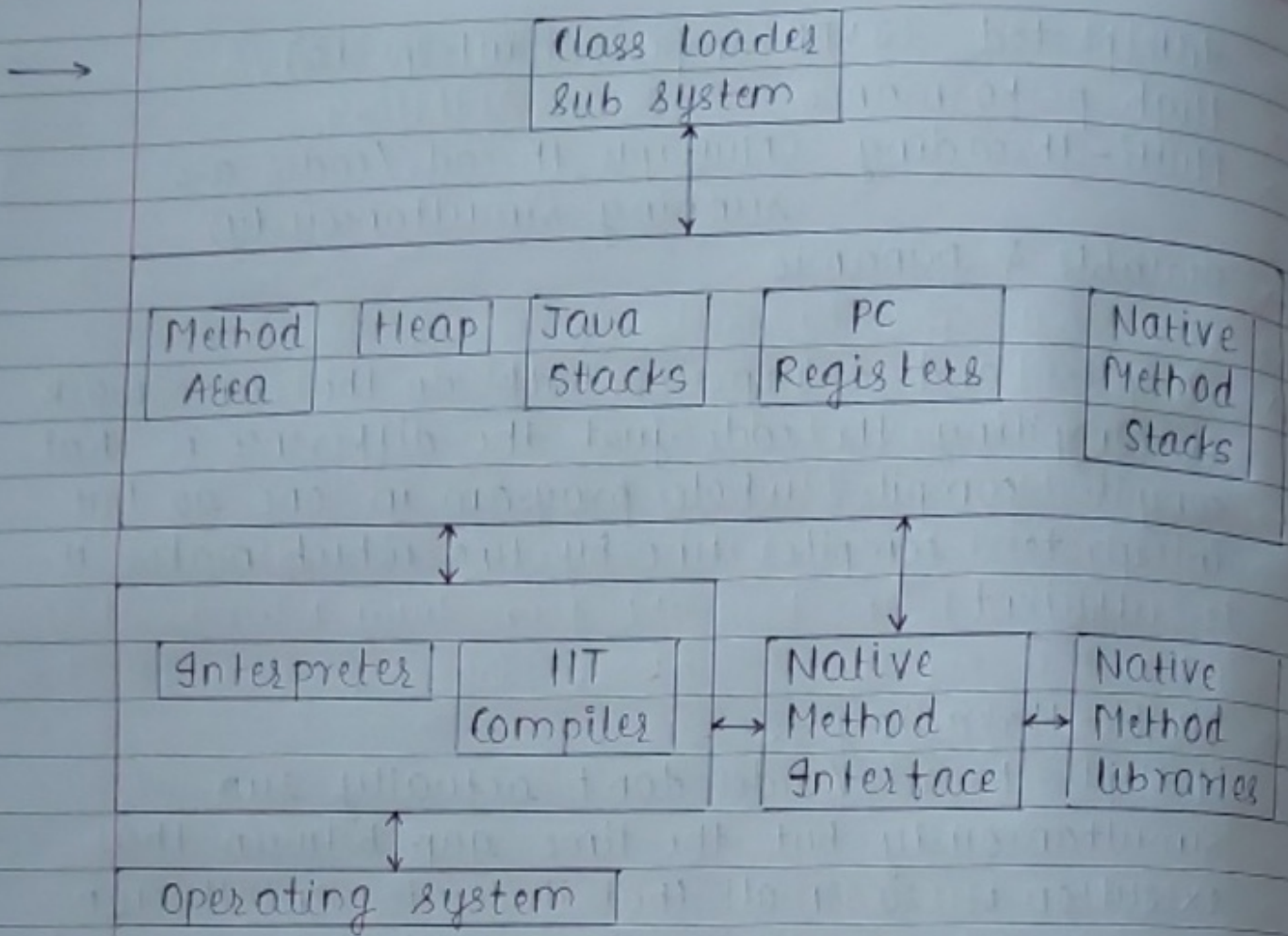
→ Garbage collector of JVM does work of destructor.
→ Java's Versions :-
Java SE (standard Edition) :-
It contains basic four Java classes used to generate standard applets & applications.
Java ME (Micro Edition) :-
Used to develop code for portable devices such as cellular phones
Java EE (Enterprise Edition) :-
Used to provide basic solutions on network.

• Features of Java
→ Simple programming language
→ Object oriented programming language
→ Distributed programming
1) Using Java we can write programs which can capture information & distribute it to clients.
→ Robust
1) Because of good exception handling mechanism
→ Secure
→ System independence

* Java :- System Independent
JVM :- System Dependent
∴ The bitcode or intermediate code is platform independent but JVM is platform dependent so bitcode runs or executes on every system which has JVM installed

→ Portability


Page No. 11
Date


Class Loader Sub System
Method Area
Heap
Java Stacks
PC Registers
Native Method Stacks
Interpreter
JIT Compiler
Native Method Interface
Native Method Libraries
Operating System