UNIT-1

Challenges in procedure oriented programming.

* In the procedure oriented programming the problem is viewed in the sequence of thing
* To solve a problem the technique of hierarchical decomposition method has been used.
* It had contains the set of instructions. for the computer organizing. which are known as functions.
* The control of flow from one action to another the flowcharts are used.

*the characteristics exhibited by procedure-oriented programming are:-

→ Emphasis is doing things (algorithms).
→ Large programs are divided into smaller programs known as functions.
→ Most of the functions share global data.
→ Data move openly around the system from function to function.
→ Functions transform data from one form to another.
→ Employs top-down approach in program design.

Introduction of object oriented design. (OOP)

* It Treats the data as a critical element in the program development
* It ties data more closely to the function. that operates on it and protect it' from outside oriented by accidental modification.
* The no. of entities of decomposition problems are known as the objectives.
* the function of one object can access the function of an another object'

Characteristics of object oriented programming.

-> Emphasis is on data rather than procedure.
-> Programs are divided into what are known as objects.
-> Data structures are designed such that they characterise the objects.
-> Functions that operate on the data of an obj are tied together in the data structure.
-> Data is hidden and cannot be accessed by external function.
-> Obj may communicate with each other through function.
-> New data & functions can be easily added whenever necessary.
-> follows bottom up approach in program design.

Procedural oriented program	object oriented program.
* In procedural programming, program is devided into small parts known as functions.	* In object orient program is divided into small parts known as object's.
* It is known as top down approach	* It is known as bottom up approach.
* No access specifier in it.	* there is accey for private, public, protected etc..
* Adding new data function is not easy.	* Adding New data function is easy.
* It doesn't have any proper way for hiding data so it is less secure.	* object oriented provides data hiding so it is more secure.
* overloading is not possible	* overloading is possible
* functions are more imp than data.	* data is more imp than functions.
* It is based on unreal world.	* It is based on real world
* Ex: C, FORTRAN, Pascal, Basic etc..	* Ex) C++, Java, python, C# etc.

Principles of the object-oriented Languages.

They are 7 types of principles they are.

1) objects
2) classes.
3) data abstraction and encapsulation.
4) Inheritance.
5) Polymorphism.
6) Dynamic binding.
7) Message passing.

objects:-

They are basic run time entities in an oop system. this program has to handle to representation of an person, a place, data, etc---. And they are also used to represent the user's - data which are known as vectors, time and lists.

* when the program is executed then the objects are interacted to send a msg to one another.
* If the message has to requested to check the account balance of an person.
* then each obj contain the data and a code.
* They don't interface b/w this without knowing the code of an object.

classes:-

* The set of data and code can be created by user with the help of class.
* objects are variables of the ~~class~~ type class.
* By example Parrot, Pegion, Crow, eagle are member of the class Birds.

Data Abstraction And Encapulation.

Encapulation is the programming Mechanism. which manipulate the code and date to binds together. when the code and data are together in an oop then the black box is created. with all necessary of data and code. Then the object is created when the code and data are together like this way.

* The attributes holds the information so they are known as Data Members.
* The functions that operate on the data are sometimes known as methods/ member function.

Polymorphism.

* Polymorphism is known as "Many forms"
* Polymorphism is an quality that allows an interface which acces a general class of actions.

Inheritance.

* The one obj is acquire the properties of an another obj this is known as Inheritance.
* It would support the classification of hierarchical
* Each obj would have to define explicitly all of it's characteristics without the useage of hierarchical classification.
* the parent can inherite it's general attribute.

Dynamic Bonding:

* The Dynamic Bonding is associated with the polymorphism and Inheritance.

* when the code is associated with a procedure of call then we can't know the time of call, until the call runs this is known as Dynamic Binding.
* Then the code makes the obj under the current reference. At run-time.
* the each class defines the object.

Message passing:-

* It is a communication b/w to with each other by sending are receiving message/ Information. by the help of "OOP".
* The sending msg is an request to an obj and receiving msg is an invocation of function.

History And Evolution of Java.

* The history of java starts with the team of java who are known as "Green team".
* The java is originally designed for interactive of an television.
* But it was advanced for the technology for the digital cable television.
* The set-top, boxes, television etc... the devices for digital devices to develop a language.
* The the netscap was incorporated the java Technology.
* simple, Robust, Portable, plat-form-independent, secured, high performance, Multithreading, Architecture Neutral, object-orient, Interpreted, and Dynamic are the principles of Java programming.

Features of Java.

The key to bring protability and security feature into a programming/computer language is the reason behaind creation of Java. And the beside of this two reasons the many reasons are moulding their imp role in language. The features are;-

1) Simple;-

The Java is easy to learn and it's syntax is simple to understand. The c++ concepts confusing and ambiguous are left out in Java. and may re implemented easily and cleanly easy to understand.

2) Object oriented.

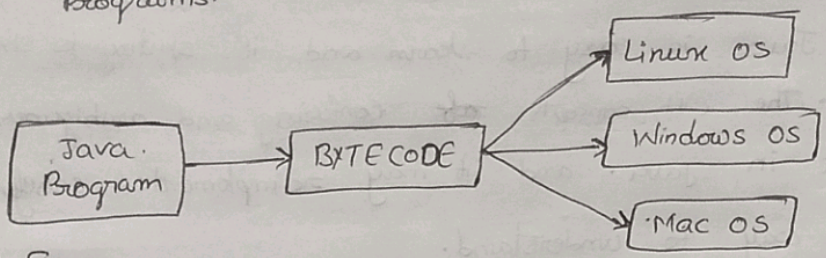
In Java very thing is based on object which had containe some data and behaviour. It can be easily extented which is based on the object model.

3) Robust;-

* Java makes an effort to eliminate error prone code by emphasizing mainly on compile time error checking and runtime checking.
* The memory of Java was improved by management and mishandled exceptions by introducing automatic Garbage collector and Exception Handling.

4) Flat-form Independent.

* C, C++ are compiled into flatform specific machines. And Java would run-anywhere when we write-once.
* this bitecode would run on any machine. And this one Provide security.
* Any machine with Java Runtime Environment can run Java Programs.



5) Secure

. the Java is more secure compare to other languages It gives secure features of virus free, temper force system.

6) Multi Threading:-

Multi threading is used to make ram do many tasks at a time. It is used to utilizes same memory and other resources to execute multi thready at a same time.

7) Architectural Neutral:-

compiler generates bytecode, which have nothing to do with a particular comp architecture, hence a Java program is easy to interpret on any machine.

8) Portable:-

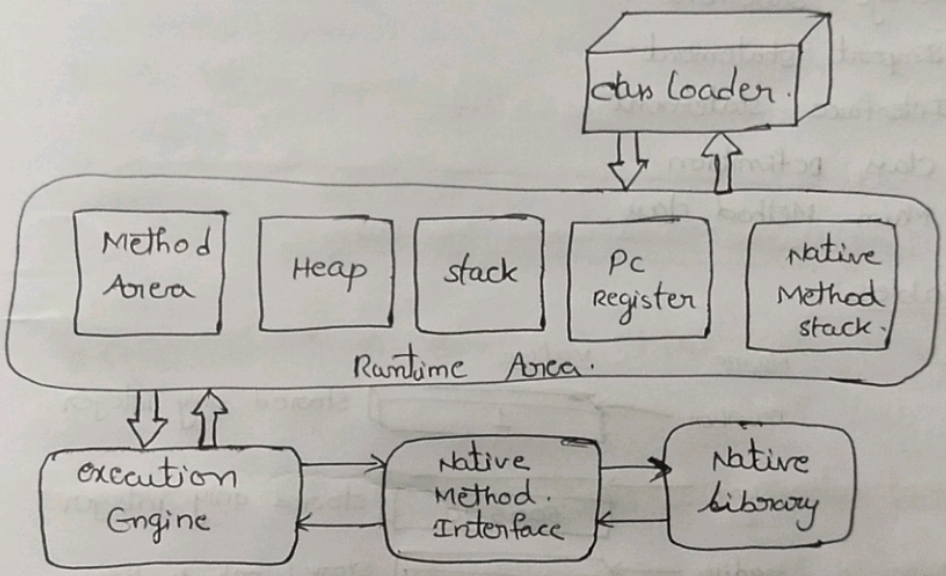
Java Byte code can be carried to any platform. No implementation dependent ~~flat~~ platform features. Everything related to storage is predefined, example:- size of primitive data type.

9) High performance:-

Java is a interpreted language, so it will never be a fast as a compiled language like "C" or C++. But, Java enable high performace with the use of just-in-time compiler.

Java Virtual Machine (JVM)

JVM is a virtual machine which provides runtime environment to execute java byte code. JVM control execution of every Java program. JVM doesn't understand Java typo, that's why we compile your *.java files to obtain *.class files. And it enables features such as automated exception handling, Garbage-collected heap.



Class Loader:- class loader loades the class for execution.

Method Area:- It stores pre-class structure as constant pool.

Heap:- It is an which object are allocated.

Stack:- The local variables and partical results are stored in stack. And the each thread contains private JVM stack. created when the thread is created.

Program Register:- It stores the address of JVM instruction where currently being executed.

Native Method stack:- It contain all nes used in application

Executive Engine:- It controls the execute of instructions which are in methods of class.

Native Method Interface:- It gives an interface b/w Java code and native code during execution.

Native Method Laibraries:- It consists of files required for the execution of native code.

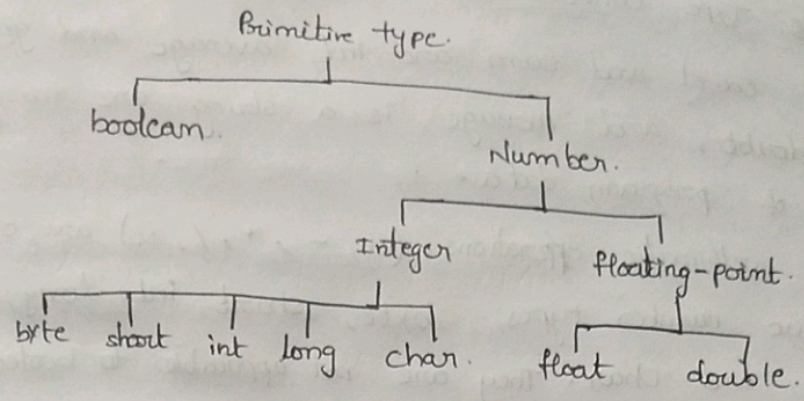
Program structure of Java.

* Documentation section.
* Package statement
* Import statement.
* Interface statement
* Class Definition.
* Main Method class.

Variables:-

Type	Name	Value	
int	number	1	stored only integer.
int	sum	500500	stored only integer.
double	radius	5.5	stored only floating-Point number.
double	area	95.0334	stored only floating-point number.
String	greeting	Hello	stored only texts.
String	statusMsg	Game over	stored only texts.

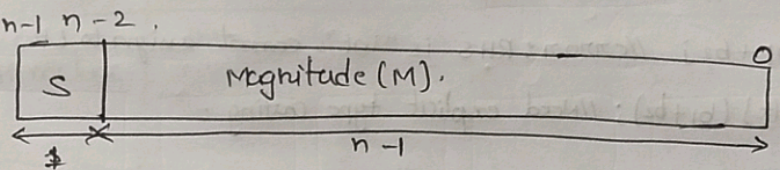
Primitive Types:-



* In asmp language, Integers and floating-point numbers are Totally Diff.

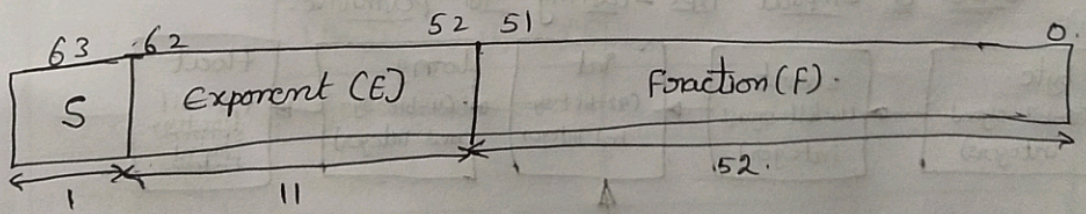
1) Integer and floating-point numbers are represented and stored differently.
2) Integer and floating-point numbers are operated differently.

n-bit integer representation.



Sign bit (s).
0 (+ve) byte (n=8), short (n=16), int (n=32), long (n=64).
1 (-ve)

64-bit Double precision floating-point Number ($F \times 2^E$)



Variable Type Conversation.

count and sum are int, average and gpa are doubt, and message is a string are many type of program data. :

* The arithmatic operations (+, -, *, /, %) are only Primitive number types;; byte, short, int, long, float, doubt, and char. They are not applicable to boolean.

* when the two operands are belongs to diff types, the value of the smaller type is automatically promoted to the larger type. And operation carried out in larger type.

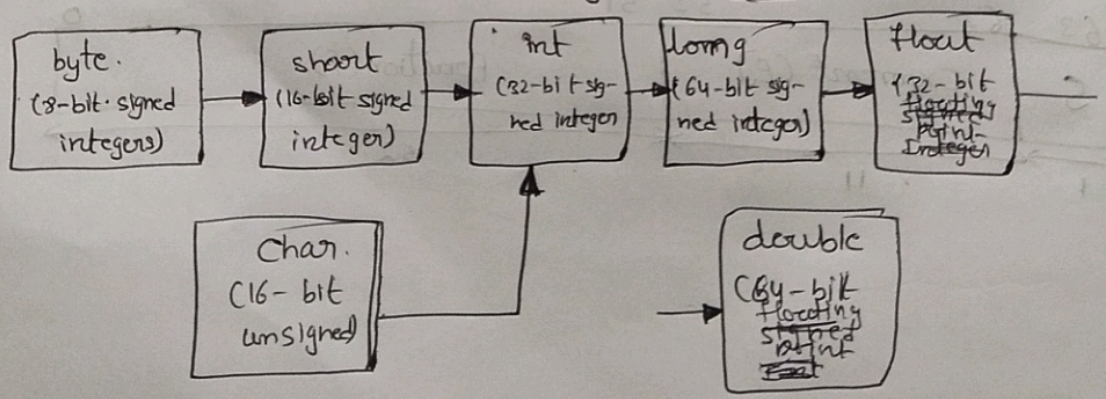
* And larger type value is evaluated.

```
byte b1 = 6,  b2 = 9  b3;
//byte + byte -> int + int -> int
b3 = b1+b2; //error: RHS is "int", cannot assign to LHS of "byte"
b3 =(byte) (b1+b2); //Need explicit type casting.
```

1) Explicit Type-Casting:- It is type of via type-casting ~~type~~

2) Implicit Type-Casting:- It performed by the compiler automatically, If there is no loss of percision.

order of Implicit Type-casting to permitive.



operators.

In Java operator is a symbol which are used to perform operations. like :- +, -, *, / etc -- .

* Unary operator.
* Arithmetic operator,
* Shift operator.
* Relational operator.
* Bitwise operator.
* Logical operator.
* Ternary operator and.
* Assignment operator.

Operator Type	Category.	Precedence.
unary	post fix.	expr++ expr--
	Pre fix.	++expr --expr +expr -expr ~ !
Arithmetic.	Multiplicative.	* / %
	additive.	+ -
Shift	shift.	<< >> >>>.
Relational	comparison.	<> <= >= = instance of
	equality.	== != .
Bitwise.	bitwise AND.	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	Logical AND	&&
	Logical OR	||
Ternary	ternary.	? :
Assignment	Assignment	=+= -= *= /= %= &= ^= |= <<= >>=

Arrays:-

* The Array is used to store a collection of data.
* It is more useful to think of an array as a collection of variables of the same type.
* Instead of using individual variables like number 0, Number 1, ... Number 99 we would declare an array like Numbe[0], Number[1], ... Numbe[99].
* They may introduce how to declare array variables, create array, and process array using indexed variable.

dataType [] arrayRefVar; //Preferred way.

Note:- The style dataType [] arrayRefVar is preferred.

String:-

It is widely used in sequence of characters, and Java programming. In Java string are treated as object.

The Java Platform provided the string class to create and manipulate strings.

String greeting = "Hello World";

String Methods