# Unit -1

* **Native code :**
  Native code is a computer programming code that is compile to run with a particular processor & its set of instructions

* **JVM :** Java Virtual Machine
  - It is a program that convert the platform generlized byte code into native code that will run on a specific processor.
  - JRE is a heart of JVM

* **Byte code :**
  ~~It is a set of instruction~~
  It is highly optimized set of instruction designed to be executed by Java Runtime (JRE) system - JVM
  - It has the advantage that it can be converted all the in to. The real machine code to run on real CPU.

* **Java Is Magic :** The Bytecode

* **Byte code characteristics**
  - Portable
  - Secure
  - Faster

→ • class file is compile version of user program & Java APIs.

* Types of loader
  - Bootstrap class loader
  - user define class loader.

* Creation of Object

```
ClassName objName; // declaration
first:
objName = new ClassName(); // initialization.
```

* Structure of Java

```
Documentation section
package statements
import statements
Inheritance Interface statement
class classname
{
    public static void main (String args[])
    {
    }
}
```

* for input
System.in
(input stream)

* for output
System.out
(output stream)

* for print (or print / println)
System.out.println(" ");
(print stream)

* for compile
javac filename

* for run
java classname
↓
(which contains main method)

→ java.lang is default package of java

* java.util
scanner is a class in java.util.
used to
take input

- Primitive → Reference data type
  known as Boxing process.

- Reference → primitive unboxing process

| String | int |
|---|---|
| NULL | float |
| Object | double |

* Convert String to Integer

```
System.out.println("Eno");
String eno = sc1.nextLine();

Int no;
no = Integer.parseInt(eno);
```

* Java use ASCII code.

= To represent NULL in java is 'null'

* Grammer : set of rules & symbol
  Lexical : How can combine to produce
  Syntatic : grammatical structure.
  Semantic : Meaning of the vocabulary symbols arranged with the structure (Structure problem)

* Character set
  - alphabet in program are known as character set
  - Java support use unicode - 32 bit

* Datatype : In java NULL represent as null
* Reference data type : Object
String
NULL
* Keyword : use for specific purpose
* Lexical element : comment, whitespace, Tokens
* Literal / constant : const datatype data
* Code convention : Not mandatory but help in prog
CLASS, METHOD, VARIABLE, CONSTANT

* Problem with nextLine :
- When you using nextInt(), float(), double(), and next() at that time it will take input repeated time and put cursor at the end. (That means \n didn't read by this methods)
- So when you are using nextLine() after any of above then it will only read a nocurring \n from buffer and it will execute same. So that it will not input from user at that time

→ Solution :
1) Use scanner sc.nextLine() before you use.
2) Use other new Object of scanner class.
3) Always use sc.nextLine() and then convert into a respected type.

* Operator & Expression
Task of comutation is known as operation
- Symbolic operator
- Named operator
- Unary operator (++, --)
- Binary operator
- Ternary operator (? :)

- Instance of operator is return boolean (T or F)
  demo. Instance of (classname)
  obj

→ Binding and binding time

  early / static / compile } compile time
  late / dynamic / execution } execution time

→ Type conversion : use for operand compatibility
→ Widening : lower rank to higher
→ narrowing : higher rank to lower

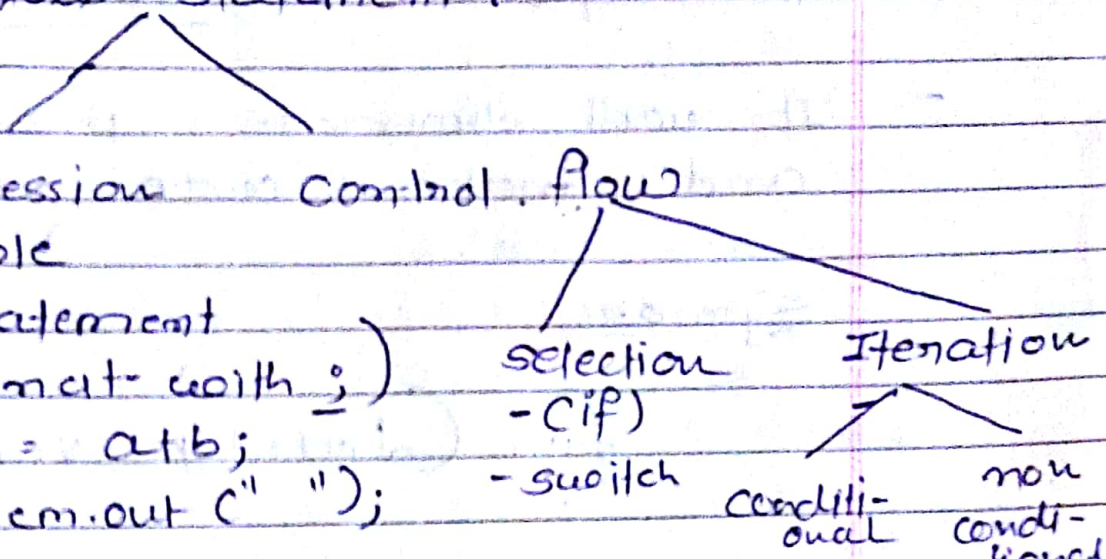
* Explicit type conversion

  Syntax : (data type) expression
  Eg: (float) 1/3 → 1.0/3.0
  (double) (10+25) → 35.000000
→ Relative
→ logical : Bitwise (&)
  Bitwise (|)
  Bitwise (^) 0^0 - 0, 1^1
→ Left shift (<<)
  left to right : 9<<2
→ right shift (>>) : 9>>2

→ Using real right shift (>>>)
  int a = -1 } right to left)
  a = a>>>24 }

→ One's complement Operator (~)
  ~2
  0000 0010
  1111 1101

* Control flow statement :




* Enum type :

```
public enum enumName
{
    enum_ele_1,
    enum_ele_2,
    :
    enum_ele_2;
}
```

* for each

```
for (int i : digit)
{
    System.out.println("Value of index " + i + "'s : " + digit[i]);
}
```

* Advantages of foreach

- It will eliminate possibilty of bugs and make code more redundant

- Syntax

```
for (datatyp varname : array)
{
    statement;
}
```

- It will print from first index of array to last one.

* Difference b/w break & break lable

(continue) (break)
|| table || continue.

```
lable :
    loop
    {
        stmt 1;
            loop 2
            {
                stmt 2;
                break lable;
            }
    }
```

We cannot use this keyword in static method.

```
class Demo
{
    p s v m ( )
    {
        Obj1 = new C1( );
        Obj2 = new C1( );

        Obj1.Chang( );
        obj2.display( );
    }
}
```

→ To call static method

Classname . methodname( );

Ex Demo . Chang( );

* Restriction for static

1) static method cannot access now static data members.

2) this and super keyword are not part of static context

3) Why we declare a main method as a static ?
— because object is not required to call static method if it were non static method.

Java program will run without main method but if and only if you have JDK < 7 version.

- JVM will create object first then call a main method, that will lead the problem of extra memory allocation.

## * Static construction / block

Syntax

```
classname
{
    static
    { ... }
    public static void main()
    { }
}
```

→ To execute before a main method (at the time of class loading) Initialization for a static variable we need to use static block.

## * Nested class

```
class outer
{
    class Inner
    { ... }
}
```

- Better visibility, performance

PAGE NO.:
DATE:

* Nested class

- Nested class are devided into,

1) static
2) Non-static

- Nested class which are declared as static are represent as static nested class.
- Non-static nested class are known as Inner class.
- A nested class is a member of its enclosing class.
- Non-static nested class has access to other member of the enclosing class, even if they are declared as a private.
- Where static nested class do not have access other member of its enclosing class.
- A member of outer class, a nested class can be declared as public, private, protected or default.

(Non Static)

Eg: class Outer
{
    class Inner
    {
        void display()
        {
            System.out.print("Nested class");
        }

PAGE NO.:
DATE:

+ → Inheritance using extends keyword

Eg: class one
{ one() }
class Two
{
Two()
{
super(); } explicitly call a constructor of base class.
system.out.print("Disp cons");
}
}

* Super keyword

To call base class constructor first in child write super(); in default constructor of child.

- It is used to call explicitly otherwise when object of child class is created base class constructor is also called.

Super keyword is like this keyword

- Super keyword should be first in construction.

- Super keyword does not have to be 1st statement in method of child class.

```
void display()
{
    System.out.print(x);
    Super.Displayone();
}

Class Base
{
    int x = 10;
}
Class child
{
    int y, x = 30;
    y = super.x;   // y = 10
```

→ Put suo;
String name;
private long mobile;
[if we want to use mobile van be another class then it will be use with setdata method.]

```
public long setmobileno()
{
    long temp = mobile;
    return temp;
}
}
```

* Diff b/w Super & this keyword

1) Use of both are same in similar context
- this is reference to the particular class
- super " " base class.
- For call super. membervar.
(base class's)

* Over riding

- same method, parameters in base and child class.
- Then it will call child class's method first and overrides the base class's method

* Over loading

- Same name.

Ex void disp (int x)
{ }
void disp (float x).