1) Why use static keyword

```
class Employee
{
  int eid;
  int sal;
  static String ceo;   // same for all employee

  public void show()
  {
    System.out.println(eid+" :"+
      salary+" ;"+ceo);
  }
}
```

creates the memory inside heap

```
Public class StaticDemo {
  Public static void main(String args[])
  {
    employee A = new employee();
    A.eid = 8;
    A.sal = 4000;
    A.ceo = "Bank"
    employee B = new employee();
    B.eid = 9;
    B.sal = 4005;
    B.ceo = "Bank";
    A.show()
    B.show()
  }
```

it assign to stack

→ so when make static it become (not an object specific)
→ so now in the heap memory ceo is not there
→ ceo is directly in class loader memory

In here when the ceo change for one object it not affect the other

So now directly use class name or else object name

ie)
Employee ceo;
(or)
e.ceo;

Note:

→ Class loader memory is in JVM there, where the class can be loaded.
→ Then only object is created in heap and assign to the stack.

Note:

So only in main method be static we need not to create object and call.

→In construct we assign value as default

```
Public Emp()
{
    eid = 1;
    Sal = 3000;
    // ceo = "Sankar";
}
```

It create assign every time when object is created

It create when object() (create can be create many times)

So In Java there is Static block.

```
Static
{
    ceo = "Sankar";
}
```

It create when class is loaded (class load only once)

now how many object can also created but only once it is executed.

Note:

→ It not depend on sequence

→ here static execute first then the construct because class can load first then only object can be created.

Note - 2

We can not use non static variable inside the Static block

eg:

```
Pu class A{
    Int a = 1;
    Public static void main(
    {
        a = 2;   ✗ ← error
    }
```

so here use Static int a = 1;

First.java during saving (3)

1) class First (Class name always starts with capital)
{
public static void main (String arg[]) — so we can use main without create an object, JVM automatically call this
{
} — To use it in outside the class
}

(this is command line arguments, it be 1, hi, hii ... any)

Note

compile: javac First.java — First.class — byte code can be created (001 ... like that)

execution: java First

First.java

2) class Demo
{
}

→ here object code we get is Demo.class not First.class

→ So in running

java Demo

---

3) Public class Demo → then must be filename same because it visible every where
{
}

---

4) When Running ie) java First

JVM can call First.main()

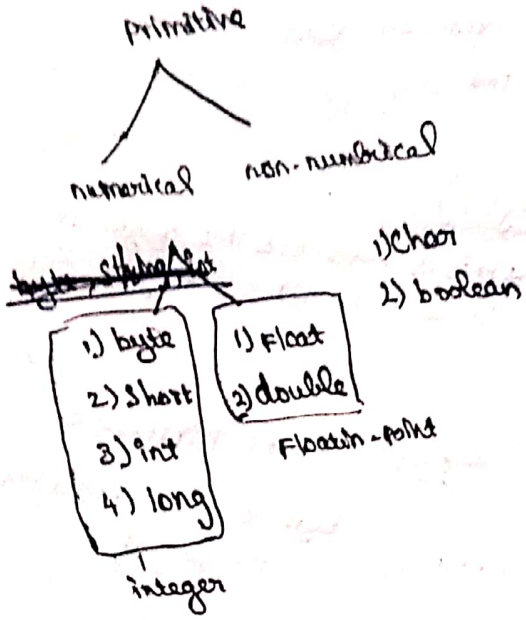
Note

> notepad

> notepad my.txt — it creates or exist it know that

> Java First hihello — running

Data types



1) byte - it takes 1 byte

1 bit for sign bit

1 - (-ive)

0 - (+ive)

for (+ive) it store directly

e.g 10

8 4 2 1

0 0 0 0 1 0 1 0

sign bit

- range

27 - (128) ie (0 to 127)

store 7 bit

27 - (128) ie (-1 to -128)

4.4.44

⇒ for send and receive transformation the purpose

Note

-ve number store in 2's compliment form

0 0 0 0 1 0 1 0

1's → 1 1 1 1 0 1 0 1

+ 1

2's → 1 1 1 1 0 1 1 0

so it become negative

→ so then add both

→ Java is 32 bit design
so it is compatable for Integer (4 byte)
int

→ long (8 byte) // if not compatable use this
|
it take 2 time process

→ short (2 byte) // some application are 16 bit already
to support that

---

floating-point (4 byte) , double (8)
|
for more acquire

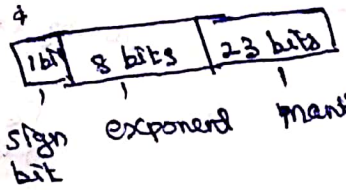
eg) 35.25

it can be store as

3525×10^{2}

3525 E2
mantisa exponent

→ we store this two here
both also can be negative
→ It convert to binary and store



Char (2 byte)
|
because
(1-byte)
of only support
ASCII

Note
here
It support
other language also
so
(unicode)

---

boolean (True / False) not 0 or 1

→ It take max of 1 byte for each boolean variable
It also be less than that that decide by JVM

Keywords:

→ directly identified by compiler

int, float, ... new ...

→ use only for that purpose

Identifiers:

variable name
class name
label
method name

---

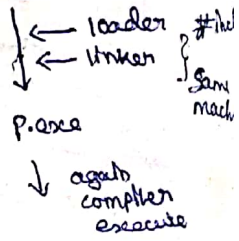
bytecode vs Object code

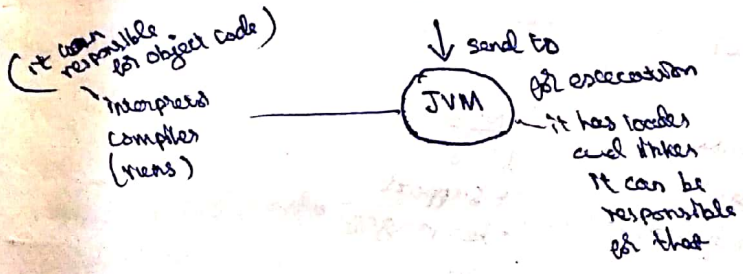
Java also high level language
↓
user understandable writing
name.java (programs

→ C, C++
after compilation
it convert to object code ie) name.obj

→ here
compilation
it to Byte code
(8 bit representation)

during running





→ So Java is Platform Independent

→ Portable (because it bytecode can execute on any platform)

System.out.println();

first take

```
class Sankar
{
  Static String s="java";
}
```

access outside by

```
Sankar.s.length();
```

Sankar → class name; s → variable Static; length() → method present in String

same way

```
System.out.println()
```

System → class name; out → Static variable

→ It present in

```
import java.lang.System;
```

System → name of class; java.lang → package

```
class System
{
  Static PrintStream out;
}
```

PrintStream → type

Note:

1) `int a=10;`
`System.out.printf("%d", a);` // likewise in c also we use here

2) In println how we can use like c.

`System.out.println(String.format("%.2f", 0.123)+"hi");`

`System.out.print(`

Getting Input

- Scanner Class
- Buffer reader class

1) Scanner – It present in `java.util.Scanner`

```
main
{
Scanner S = new Scanner (System.in);
```

- `S` : assign
- `new Scanner (System.in)` : it take memory create in heap
- `System.in` : InputStream
- `S` : in stack

```
int a = S.nextInt ();
float b = S.nextFloat();
String c = S.nextLine();
char d = S.next.charAt(0);
}
```

Note

- System.in = Read data from keyboard
- System.out = display the output
- System.err = Display the error

Input streams

bytes of data

These are present in `io.lang` packages

Here notes:

→ If we get a Int float any first and get the string or char it will be terminate.

→ because that enter (newline) also take as input to that char so we give

```
main()
{
Scanner S = new Scanner(System.in)
    int a = S.nextInt();
    char b = S.next.charAt(0);
}
```

→ `S.nextLine()` — So we give like this

so move to :

Buffer reader (io. Package)

It can read data both from file and keyboard

```
import java.io.BufferedReader
import java.io.InputStreamReader

main()
{
InputStreamReader ip = new InputStreamReader(
                                System.in);

BufferedReader br = new BufferedReader(ip);

String a = br.readLine();
int b = Integer.parseFloat(br.readLine());

float f = Float.parseFloat( " " );
}
```

It convert string input to float here like wise.

Note

→ BufferReader only work on character stream

→ But our i/p is on bit stream

→ so need to conver so we use InputStream Reader.

Looping

same

- extra foreach loop is there

i) Array declaration

- int a[] = new int[10];
- " []a = " "

working of foreach

```
for(int i:a)
 {
   System.out.println( i );
 }
```

it print the values

Note

if need to use array methods like

Sort()

it available in Array.

i.e) Util.Array package

* Array.sort(a);

Note:

In output println leave space both side

—

→ Print here

—

- Conditional statement same
- Switch is same

Wrapper class

here every datatype there is a class

like int - Integer class

float - Float class

```
main
{
 int i = 5;
 Integer i1 = new Integer(5);        — this is called Boxing
                                     — it is best because of object oriented.
 Integer i2 = 5   } auto boxing
          or
          i

 int j = i1.intValue(); // unboxing

 int k = i1; autounboxing

}
```

Functions also same

```
Public class A
    {
     void Bn()
     {
     }
     A a = new A()
       a.Bn()
}
```

is static → B.B()

```
Class B
{
 A a = new A()
 a.Bn()
```

→ A.Bn()

Class

→ Defines the structure and working of object

Object - real instance

→ It is used to access the class

→ every obj has behaviour (wt knows, wt does)

OR that we use class as blueprint

Source code - we write

• .java

↓ compiled

bytecode

• .class } all OS can understand

developed

→ James gosling in 1995

→ The name from Java Coffee.

→ oracle

→ In single package there may be many class

JDK - Java Development Kit (all tools) — develop, compile and run

IDE - Integrated Development Environment (where we can code, only run)

JRE - Java Runtime Environment

JVM - Java Virtual Machine

Some IDEs + It is part of JRE

Convert byte to object code

Interpret to execute java program (byte code) like by the

10110

Note

class name as the file name

Scanned with CamScanner