7

Java :-

⇒ It is object oriented Programming language.
⇒ It is technology.
⇒ It is fully case sensitive language.

Types of Java :-

(i) JSE (Java standard Edition) (core Java)
Window based like college management system store management.

(ii) JEE (Java Enterprise Edition) (Advance Java)
(Web based app like irctc, fb chat module, linkedin etc.

(iii) JME (Java micro edition) (Android Java)
Mobile based application

(iv) Java with embedded

(v) Java cards.
NFC - Near Filled Communication sensor use in metro cards and wifi atm cards.

Note :- Small Talk is 1st object oriented Programming language but everyone knows about C++ (1980).

Problems of C & C++.

(i) No security
⇒ easily hackable
⇒ due to pointer | store same memory address add multiple place.

(ii) Plateform dependency

Windows OS extension ⇒ .exe
mac OS " ⇒ .dmg
Linux OS " ⇒ .rpm

(iii) Reusibility (but partially solved by C++)

Note :- #getch() :- it is used to get single Character input.
In notepad if we save the C program use .c.

Software :- Collection of multiple program.

Types of software :-
(1) System software :- It is general-purpose software that manages basic system resources and processors.
Ex:- Mac, Linux, Android, Microsoft Windows, game engines, search engines.

(2) Application software :-

SUN MICRO SYSTEM (USA) company takeover the Green Project (Problems in C/c++) by James Gosling & Team completed in 18 months.

They completed the Project work in 1991 as name of OAK (Name of tree) but it was earlier registered but they choose the name JAVA (they drink coffee that's why the name of coffee chosen as JAVA Programming language with symbol of coffee.

⇒ Later, Java was bought by company ORACLE, With same symbol and name.

⇒ Java 1st version was launched in 1996 named as JDK 1.0 (Java Development Kit)

Q What is JAVA JDK 1.0?
It is a collection of Java library and Java tools.

⇒ Java updated version is JDK 6, JDK 7, JDK 8 - etc
⇒ All others are updated version.
⇒ Mostly everyone use JDK 8.

STEPS TO CONFIGURE JAVA IN OUR WINDOWS

1. First install jdk in our laptop.
2. Open c drive then open program files and double click on JAVA directory.
3. Open jdk directory and then open bin folder.
4. Copy the current path
   Ex. → C:\Program Files\Java\jdk-16.0.2\bin
5. Right click on my computer icon. Select properties option.
6. Click on Advance system settings.
7. Click on environment variable button.
8. Select Path variable and then click on edit button.
9. Click on New button and Paste your current Path.
10. Now click ok button at three times.
11. Now Press window + R, then search cmd.
12. Now write java -version, then press enter.

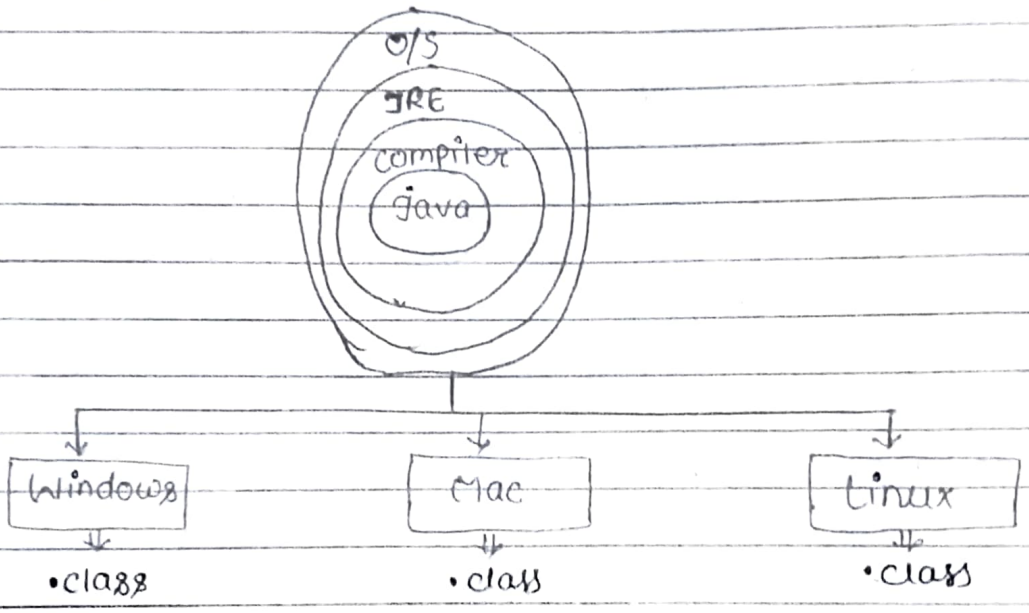
Basic commands of cmd :-

For Change the drive — drivename : enter
For Change the directory — cd directory name ent
For clear screen — cls
For back directory — cd.. enter
For compile java Program — javac filename.java enter (java compiler name is javac)
For Run the program — java classname enter

Features of Java

(1) Plateform independent
(2) Secure
(3) Robust (Strong)
(4) Simple
(5) Poartable
(6) open source

(1) Plateform independent :-



Note :- Any Programming Language are called Plateform independent if that Programming Language are supports CORA feature.

compile once Run Anywhere

⇒ Java Plateform Independent due to JRE.

(2) Secure :-

⇒ It is not support the pointer.

⇒ It's Byte code.

Bytecode :- Bytecode is the 16 bit unique and universal set of optimization code.

⇒ This code are developed with the help of unique code system.

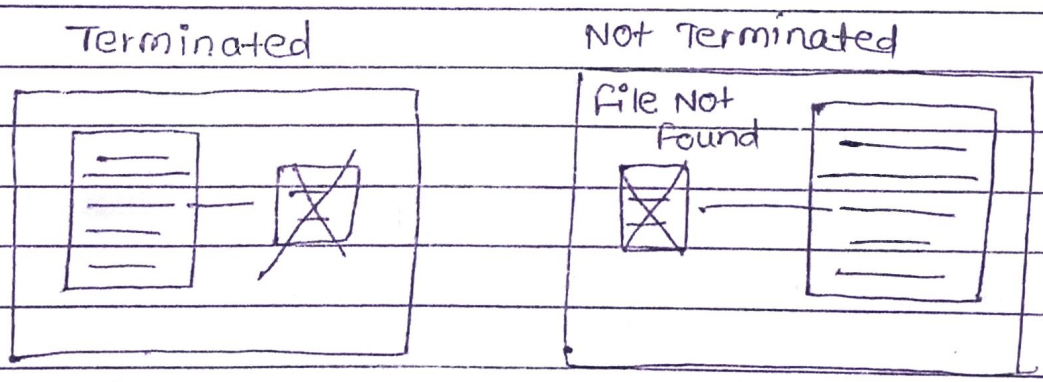
⇒ you can not edit or customize your bytecode.

⇒ Byte code are exists in .class file.

Note :- If u want to view discription of .class file so you have to use java O compiler.

27-09-21
Monday

(3) Robust :- (Strong)



Error

Exception Handling

NOTE :- Java is strong Programming language due to Exception Handling concept.

(4) Portable :-

Any Programming language called Portable Programming language is this Programming language are support WORA feature.

↓ WRITE ONCE RUN ANYWHERE

(5) Simple :-

⇒ It's support the concept of OOPS.

⇒ It's automatic Garbage collector. (during Run time)

(It helps in clean unused memory)

NOTE :- From JDK 1.5, we have a capability to call garbage collector manually using of finalized function.

(6) Open source :-

⇒ Free of cost

⇒ Public

First Java Program

```
Class Ainwik
{
Public static void main (String St[])
{
System.out.Println("Welcome To The World of Technology!");
}
}
```

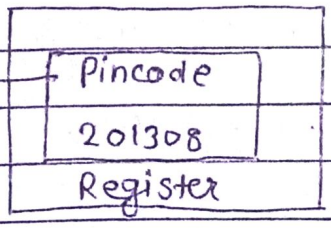
-09-21
Tuesday

Basic Fundamental for develop a program in any Programming language:-

1. Variable or data-member
2. Data-type
3. Access specifier
4. Modifier
5. Method or Member-function
6. Identifier
7. operators
8. Conditional statement
   (if, else, else if, nested if)
9. Loop
   (for, while, do-while, foreach)
10. Switch statement
11. Keywords
12. Comments

(1.) Variable or data-member

Variable are nothing it's just a name of memory location where exist our data.



Types Of variable :-

1. Instance variable or non-static variable
2. Local variable
3. Static variable

(1) Instance variable or non-static variable :-

Instance variable are always define within class but outside of any function, block or constructor or methods.

⇒ Instance variable are acessable from anywhere within class.

NOTE :- If you want to intialize or call instance variable so you must be create and Object.

Program :-

```
Class Aiwik
{
  int Pincode = 201308; // this is instance
                           variable
  Public Static void main (String St[])
  {
    Aiwik obj = new Aiwik();

System.out.println("your Pincode is :-
                  " +obj.Pincode);
  }
}
```

NOTE :- If you don't provide any value in instance variable so by default our compiler will be

(2) ~~Local~~

Syntax to declare variable :-

1. data_type var-name;
2. data_type var-name = value;
3. data_type var1, var2, var3;

Ex :- 1. int x;
2. int x = 10;
3. int x, y, z;

(2) Local variable :-

Local variable are always define within any function or any block or any constructor

⇒ you can not access local variable outside of any method or any block or any constructor.

Imp Note :- If you want to use local variable in our program, so you must be initialize it because there are no any concept of default value.

(3) Static variable :-

Data-type

Data types are defines the two things of the data

(i) Types of the data

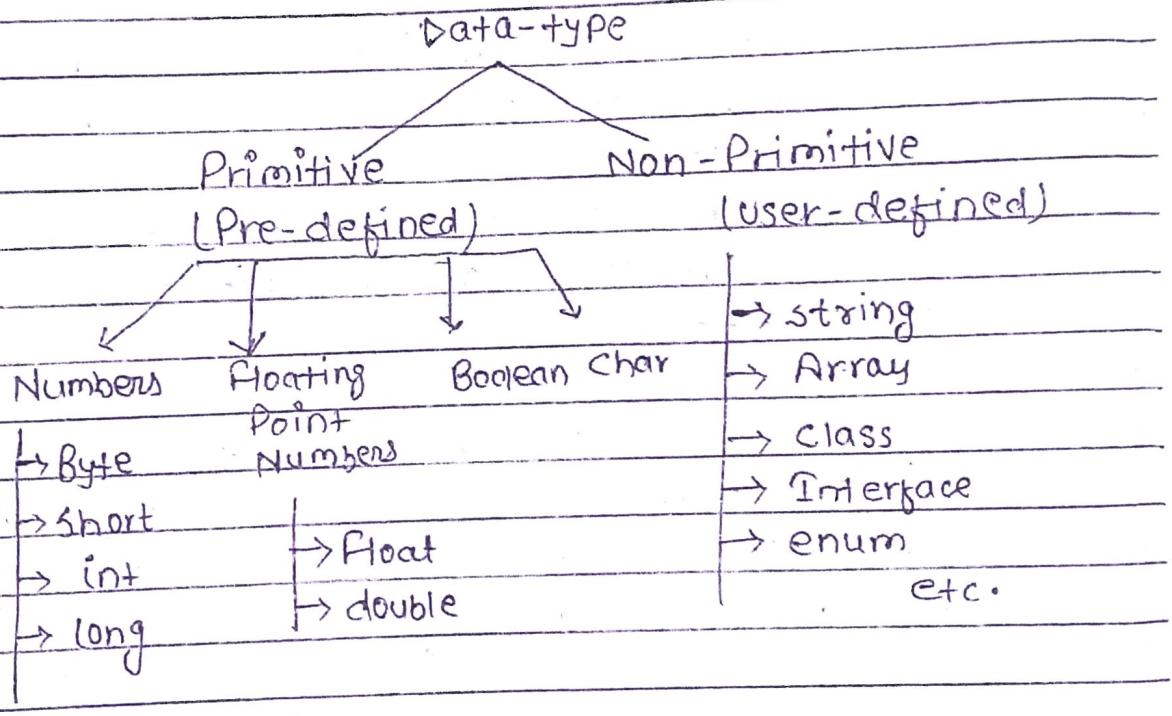
(ii) Size of the data

Ex:- type Numbers

int x = 123

size 4 byte

Types of data-type :-



Primitive data-type

data-type	size	default value	Range
byte	1 byte	0	-128 to +127
Short	2 byte	0	
int	4 byte	0	
long	8 byte	0	
Float	4 byte	0.0	
double	8 byte	0.0	
char	2 byte	(\u0000)	
boolean	1 bit	false	

Formula for Range

$$-2^{n-1} \text{ to } +2^{n-1}-1$$

where n = no. of bits

Range of byte

-2^{8-1} to $2^{8-1}-1$

-2^{7} to $2^{7}-1$

-128 to +127

30-sep-21
Thursday

NOTE:-

There are no any default size and no any default value of any non-primitive datatypes.

(3) Access specifier :-

Actual use in Package

Access specifier are used to specify the scope and accessibility of class variable and functions.

Access specifier	class	variable	method
Public	true	true	true
Protected	false	true	true
Private	false	true	true
default	-	-	true (in case of interface)

(4) Modifier :-

Modifier are used to change the behaviour and functionality of class, variable and methods.

Modifier	class	variable	Method
final	true	true	true
static	false	True	True
abstract	true	false	true
synchronize	false	false	true

2021
Friday

15) Method or Member - function :-

If you want to achelve any task in any Programming languages so you must be create method) define a function.

Syntax to define a function :-

optional

Access specifier + modifier + return - type + function - name (arg if any)

{

}

Eg :- Public Final void disp()

{

}

or

Public Final int disp()

{

return 0;

}

Program :-

Class Ainwik

{

Public Static void disp()

{

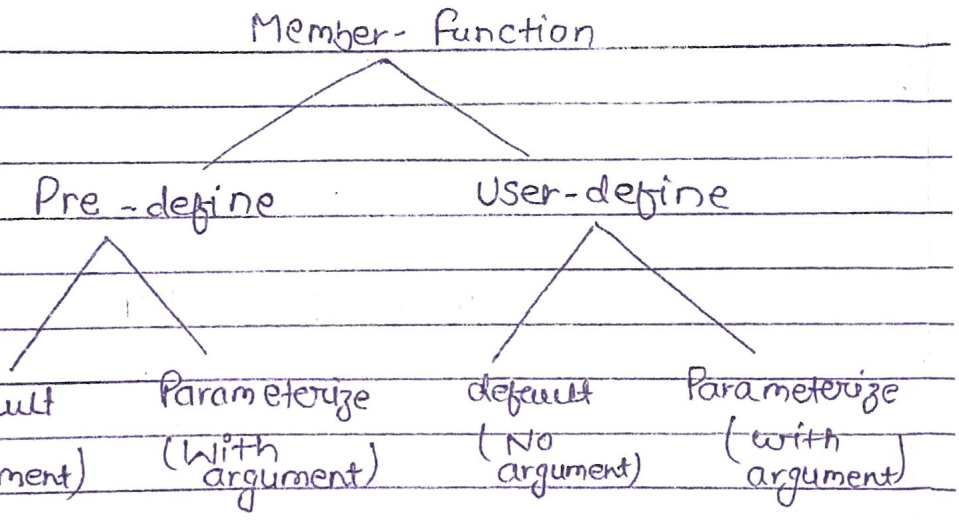
```
System.out.Println("Hello class!");
}

Public Static void main(String arg[])
{
   disp();
}
}
```

Types of method:-

There are two types of methods

(i) Pre-define

(ii) User-define



Program

```
class Ainwik
{
Public static void disp() // no argument
```

```
{
System.out.Println("Welcome to Ainwik
                    Infotech!");
}
Public static void sum (int x, int y)
                    // with argument function
{
  System.out.Println("sum of two no:-
                    " +(x+y));
}
  Public static void main (String arg[])
{
  disp();
  Sum (10,2);
}
}
```

NOTE:- If you want to provide same output for different different user, so we have to use default function and if you want to provide user specific response (Output) so you have to use parameterize function.

Assignment :-

Write a program to perform arithmatic operations like Addition, Substraction, Multiplication and division.

Identifier:-

Identifier are used to identify the name of class, the name of variable, method, function also.

```
Class <class- identifier>
{
data _type <variable- identifier>

return _type <function - identifier>()
{ }

}
```

Rules to declare class identifiers:-

(1) The first letter of the class name should be capital letter.

(2) Numeric value are allowed but don't write as the first letter as a class- name.

(3) Special symbols are not allowed.

(4) Space are not allowed.

(5) Underscore are allowed.

(6) Don't write keywords name as a class-name

Rule to declare variable identifier :-

(1) The first letter of the variable name should be small letter.

(2) Second - connecting word letter is capital letter. (If is also for function).

(8)

Q Operators :-
It is used to perform some operations between two or more operands.

Types of Operators:
(i) Arithmatic operator
(ii) Assignment ,,
(iii) Relational ,,
(iv) logical ,,
(v) conditional or MISC ,,

Date 5/10/21 Tuesday

(1) Arithmatic operator :-
It is used to perform some arithmatical operation between two or more operands.

(+ - * / % ++ --)

% ⇓ Remainder
++ ⇓ increment
-- ⇓ Decrement

Program :-

```
class Ainwik
{
Public static void main (String s+[])
{
    int x = 6;
    int y = 4;
    System.out.Println (x+y);
      ,,    ,,    ,,   (x-y);
      ,,    ,,    ,,   (x*y);
      ,,    ,,    ,,   (x/y);
      ,,    ,,    ,,   (x%y);
}
}
```

Output:

10
2
24
1
2

* Increment (++):-

(i) Post increment (x++):-

Rule:- Assign first then increment

(ii) Pre increment (++x):-

Rule :- Increment first then assign

Program

① // Post Increment

```
Class Ainwik
{
 Public static void main (String St[])
{
    int x=0, y=0;
      y= x++;

    System.out.println (y);
}
}
```

Output:-

0

② // Post Increment

```
Class Ainwik
{
Public static void main(String St[])
{
    int x=0, y=0;
      y=x++;
    System.out.Println (y++);
    system.out.Println (y++);
    System.out.Println (y);
}
}
```

Output

0
1
2

③ // Post increment

```
class Ainwik
{
Public static void main (String St[])
{
    int x=0;
      x++;
   System.out.Println (x);
}
}
```

Output
1

④ // Pre increment

```
class Ainwik
{
Public static void main (String St[])
{
    int x=0;
     ++x;
  System.out.Println(x);
}
}
```

Output
1

⑤

```
int x=0, y=0
    y=++x;
S.O.P(y);
```

output = 1

Note :-

x = 0;	x = 0;
x = x + 1;	x++;
System.out.Println(x)	S.O.P(x);
output = 1	output = 1

use x++ over x=x+1 as it reduce time taken (time complexity) as it has only one opererator '++' & other has two "=" & "+".

(2) Assignment operator :-

Assignment operator are used to assign a value into single operands.

⇒ Assignment operator is always perform operation from Right to left side.

⇒ =, +=, -=, *=, /=, %=

Program

```
Class Assignment
{
Public Static void main (String St[])
{
    int x = 10;
      x /= 2;
  System.out.Println(x);
}
```

Output :-

(3) Relational operator :-

Relational operator are used to check the relation or find out the condition between two or more operands.

NOTE :- Relational Operator are always return the boolean type result, it means it will be return true or false.

Program

```
// Relational operator
class Ainwik
{
Public static void main (String St[])
{
int x=10;
int y = 20;
System.out.Println(x>y); // greater than
//  //  //  (x<y); // less than
//  //  //  (x≥y); // Greater than equals to
//  //  //  (x≤y); // less than equals to
//  //  //  (x=y); // equals to
//  //  //  (x!=y); // Not equal to
}
}
```

Output

```
False
True
False
True
False
```

(4) Logical Operator :-

If you want to check two one or more conditions in a single line so you have to ~~call~~ use logical operators.

&& (logical AND)
|| (logical OR)
! (logical NOT)

(1) && (logical AND)

case 1 :- con1 && con2 && con3 → False
↓ true, ↓ true, ↓ False

case 2 :- con1 && con2 && con3 → True
↓ true, ↓ true, ↓ true

Program :-

```
class Ainwik
{
Public static void main(String st[])
{
int x=10;
int y=20;
int z=25;
System.out.Println(x>y && x<y);
"  "  "  (x<y && x>z);
"  "  "  (x<y && x<z);
}
```

Output

In case of logical operator all conditions must be true if any one condition going to false so it will be return false result.

(2) || (Logical OR) :-

case 1 :- con1 || con2 || con3

con1	con2	con3	
True	False	True	True
False	False	False	False
True	True	True	True

(3) ~~%~~ (Logical

Program :-

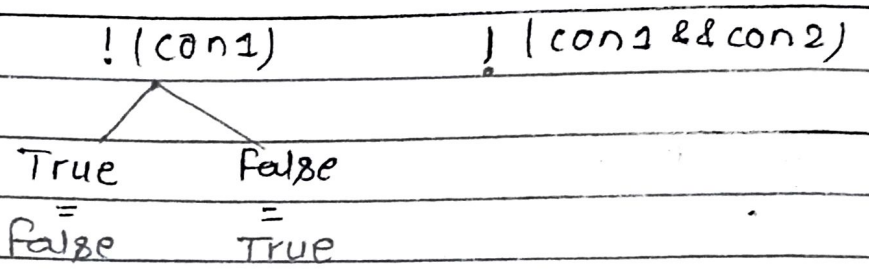
```
Class Ainalik
{
Public Static void main (String st[])
{
    int x=20;
    int y=10;
    int z=30;
System.out.Println (x>y || x<y);
System.out.Println (x<y || x<3);
System.out.Println (x>3 || y>3);
}
}
```

Output :-

True
True
False

(3) Logical NOT (!)

Logical NOT operator are treated as a game 'changer' it means when your result is true so in this case it will be return the false result and when your result is false so it will be return the true.



(5) ~~CONDITIONAL OPERATOR~~ :-

```
// Logical NOT operator
Class Aincrik {
Public static void main (string st [])
{

int x=10;
int y=20;
int z = 25;
System.out.Println (!x>y || x<y);

}
}
```

Output :-

False

15) Conditional operator:-

It is use to reduce the functionality of if & else statement, it means if you want to use single line statement in if & else concept so you must be need to use conditional operator you don't write if else statement.

Syntax:-

data_type variable_name = (condition)?
Statement if true : Statement if false
(true) (false)

Ex:- String result = (marks > 40)? "PASS" : "FAIL";

Program:-

```
// conditional operater
class conditional
{
Public static void main (String St[])
{
  int age=10;
String result = (age >=18)? "you are valid for vote" : "Beta Ghar Jao";
  }
}
```

Output:-

Beta Ghar jao

* Operator over loading :-

In java their are no any concept of operator overloading.

But one operator are by default overloaded in java that's name is "+" operator.

Program :-

```
class Ainwik
{
  Public static void main (String St[])
{
  int x=10;
System.out.Println ("Hello" +x);
  }
}
```

Output :-

Hello 10