**09/03/2024**

Client -> Make Request (Zoom Application will be the Client)

Server -> where we are getting the response

Code -> Developers

Build -> Compress and Compile Code -> Tester

Application -> Used by the Customers -> Customers

**09/04/2024**

* Project -> Start with Capital
  + Package -> Start with lower case -> can have.
    - Class -> Start with Capital

Upper case -> ARUN

Lower case -> arun

Camel Case -> arunKumar

Tokens in Java (Smallest Unit of any Program)

(IK LS CO)

* I -> Identifier (Name given to the ClassName) EX: Addition, Calculator
  + Rules of Identifier
    - It should always starts with Capital Letter
    - We can use underscore
    - Can be AlphaNumeric
    - No Special Char other than under Score
    - Should not be Numeric
    - Should not be Numeric + alpha
* K -> Keywords ->
* L -> Literals
* C -> Comments
* O -> Operator

**09/05/2024**

* K -> Keywords -> Reserved Word in Java and there are totally 50 Keywords
  + Entire Keywords are in Lower Class
  + Each keyword has meaning in Java
    - Abstract
    - Assert
    - Boolean
    - Integer
    - Character
    - Final
    - Finally
    - Static
    - Break
* L -> Literals -> Values used in program Called as Literals
  + DOB, Age, Name, Father Name
* Separators -> Separate the Code from each other
  + , -> Comma
  + ; -> Semi colon
  + [] -> Bracket
  + {} -> Paranthesis
  + () ->
* C -> Comments
  + We are not deleting Just we are Ignoring
  + Type
    - Single Line Comment (//)
    - Multi Line Comment (/\* \*/)
* O -> Operator
  + +
  + –
  + \*
  + /
  + % -> Modulus
  + ++ -> Increment Operator
  + –- -> Decrement Operator
  + || -> OR Operator
  + && -> AND Operator
  + = -> Assigment Operator
  + > -> Greater than
  + < -> Less than
  + <= -> Greater than or Equals to
  + >= -> Lesser than or Equals to
  + == -> Equals to Equals to
  + ! -> Not Operator
  + != -> Not Equal

System.out.print**ln** -> Moves the Control to Next Line

System.out.print -> Not Moves the Control to Next Line

**How Java Program works**

Compilation Interpretation

Binary

* Read Line by Line
* Execute
* translate
* Syntax
* Rules
* Translate

.java .class OS

Compile time error Run time Error Output

Exception

**09/06/2024**

JDK – Java Development KIT -> Will have the Library Files will be Needed to write the Program

JRE – Java Run time Environment -> It Creates the environment where you can run the Program

Java is platform independent

JVM – Java Virtual Machine -> Only Responsible for Executing the Program Line By Line Starting from main Method

JIT – Just In Time -> Converting .class into Binary

What is main method ?

Public Static void main (String[] args)

* public -> Access Specifier
* static -> Modifier
* void -> return Type
* main -> Method Name
* String[] args -> Parameter / Argument

**Access Specifier**

* public
* protected
* package / Default (if we are not specific any access specifier then by default it will assign default)
* private

**Modifier**

* Static Modifier
* Non Static Modifier (No need to specify the Modifier for Non Static)

**Return Type**

* byte
* int
* char
* long
* float
* String -> Class so it is Uppercase
* void
* Object -> Class so it is Uppercase
* **Return type of main method is void**

**Method Names**

Method Name can be any like name, addition, Login

How to call static method as Main Method

Assignment 1:

Call 5 different Static method call that inside main method

**09/09/2024**

How to call Non Static method in Main Method

Procedure:

1. Create an Object
   1. Syntax : classname refvariable = new Classname
2. Need to call Non Static Method with help of Reference variable

In a class can we have both Non-Static and Static method: Yes

How to Call the parameterized static method inside main method

Data Types and variables in Java:

* byte -
* short Numbers (Different Range of Data) -
* int
* long
* float
* double Decimals
* char Alphabets
* Boolean True Or False
* String

Formula to calculate data type : -2n-1 to 2n-1 – 1

n – size of bits

1TB – 1024 GB

1GB – 1024 MB

1MB – 1024 KB

1KB – 1024 Bytes

1Bytes – 8 Bits

|  |  |  |  |
| --- | --- | --- | --- |
|  | Byte | Bits | 2n-1 to 2n-1 – 1 |
| Byte | 1 | 8 | -128 to 127 |
| Short | 2 | 16 | -32768 to 32767 |
| Int | 4 | 32 |  |
| Long | 8 | 64 |  |

Syntax of Datatype and Variable in Java

datatype variable = value;

example : int a = 10;

float a1 = 2.54f

**09/10/2024**

Variables: Container to Store value

Literals: Values which we are storing in Variables

**Operators**

* + - \* / %

**Method Overloading**

* Developing Multiple method with same Name but different parameter or Argument list

**Constructor:**

* Special type of method in java
* Does not have return type
* Can have any type of access specifier
* Constructor Name will be same as class name
* Constructor will always be Non-Static
* Constructor will be called by creating the Object

**Syntax:**

class One {

One() { // Constructor

}

**09/11/2024**

Constructor Overloading -> Developing Multiple Constructor with same Class name and different arguments

Can we Overload Main Method 🡪 Yes

How to accept the Human Input at run time -> Generic way

Scanner Class -> to write the program in generic program

* Its Predefined Class in Java
* It Present in java.util package
  + **Scanner s1 = new Scanner (System.in);**
  + System is the class that comes from java.lang Package
  + System.in Means accepting input
  + sysout.out means giving output
* Methods of Scanner Class -> Method used to accept the Human input
  + s1.next();
  + s1.nextByte()
  + s1.nextInt()
  + s1.nextShort()
  + s1.nextLong()
  + s1.nextFloat()
  + s1.nextDouble()
  + s1.nextBoolean()

**09/12/2024**

* How to do initialization and declartion using Scanner class

Declaration and Initialization using scanner class

* Input mismatch Exception
  + nextInt if we are passing String or Double

**byte, short, int, long is different in String**

**byte, short, int, long, Boolean -> Keywords is different in**

**String -> Class (Java.lang)**

**System -> class (Java.lang)**

package we are utilize by importing

java.lang no need to import

Cf means final class

Types of Variables:

Container to store the value

* Local Variable
  + Any Variable which is declared inside the method is called as Local Variable
  + Scope of Local Variable is from the beginning of the method to end of Method
  + Local Variable can never be distinguished between static and Non-Static
  + Local variable will never have the default value
  + Local Variable can be utilized based on the declaration and initialized (Without initialization and Declaration we can use the local variable)
* Global Variable
  + Any Variable which is declared inside the class is called as Local Variable
  + Scope of Local Variable is from the beginning of the class to end of class
  + Global Variable can be distinguished between static and Non-Static
  + Global variable will have the default value
  + Global variable can be utilized until even without initializing it

**09/13/2024**

* What will be the default value of each datatype when it is declared as Global Variable

|  |  |
| --- | --- |
| Double | "0.0" |
| Byte | 0 |
| Short | 0 |
| Int | 0 |
| Long | 0 |
| float | ""0.0 |
| Char | not defined |
| boolean | FALSE |
| String | null |

* Global Variable can’t be updated Globally

Final Variable:

It will not be changing kind of scientific truth

Syntax: final int a= 10;

Both Global and Local can be final

**Arithmetic Operators:**

Increment Operator (i++)

Decrement operator(i--)

**Comparison Operator:**

* < Less than
* > Greater than
* <= lessthan or equal
* >= Greaterthan or equal
* == equal to equals to
* != not equals

What is difference between = and ==

Int a = 10: Assignment Operators

“==” => Comparison Operators

**Logical Operators:**

* + Or
  + And
  + Not

**Conditional Statement:**

* If block -
* If else block – 1 Block
* If else if block – 1 Block
* Nested if else block

**09/18/2024**

Logical Operators

* AND
  + Syntax: if(C1 && C2){

}

* OR
  + Syntax: if(C1 || C2){

}

Table for AND Operator

|  |  |  |
| --- | --- | --- |
| c1 | c2 | Output |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE |

Table for OR Operator

|  |  |  |
| --- | --- | --- |
| c1 | c2 | Output |
| TRUE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

Table for NOT Operator

|  |  |
| --- | --- |
| c1 | output |
| TRUE | FALSE |
| FALSE | TRUE |

**Both And and OR operator will be used along with NOT Operators**

* + Syntax : if(!(C1 && C2)){

}

* + Syntax : if(!(C1 || C2)){

}

How to execute same things Multiple times

**Looping Concept:**

* for
  + Syntax:
    - for (declaration and initialization; condition; increment or decrement operator) {

logic  
}

ex:

for (int i=1;i<=10;i++){

}

* while
* for each
* do while

Print 1 to 10 using for Loop

**09/19/2024**

In For Loop How to Print 10 to 1

Math class in Java:

* has only Static method
* Methods in Math Class
  + addexact(int x, int y)
  + addexact(long x, long y)
  + subtractexact(int x, int y)
  + subtractexact(long x, long y)
  + multiplytexact(int x, int y)
  + multiplytexact( x, int y)
  + multiplytexact(long x, long y)
  + max(double x,double y)

**Thread**

Method: Thread.sleep() -> make the program to stop for some time

**09/20/2024**

Keyword

* continue -> can be used to Skip the particular iteration
* assert -> it will check the condition (we need to enable by Runas -> Argument -> -ea)
* switch ->
  + Syntax: switch(1)

{

case 1: action 1

break;

case 2: action 2

break;

case 3: action 3

break;

default:

}

* break
* case
* default

How to run the java program from cmd prompt

Compilation: javac Filename.java

Interpretation: java Filename.java

09/24/2022

Inheritance -> Property extends and implements

* Single
* Multi-level
* Multiple level -> it is not possible to achieve through the class in Java but it can be achieved through the interface
* Hierarchical Level -> Opposite of Multiple level
* Hybrid Level – Combination of Single Level, Multi-Level, Hierarchical level In heritance
* Never able to inherit the Constructor

Super Calling Statement: -> to call parent class constructor from the child class constructor

Super calling statement will be always the first line

They can be written in 2 ways

Implicitly ->No need to specify

Explicitly -> we can mention it

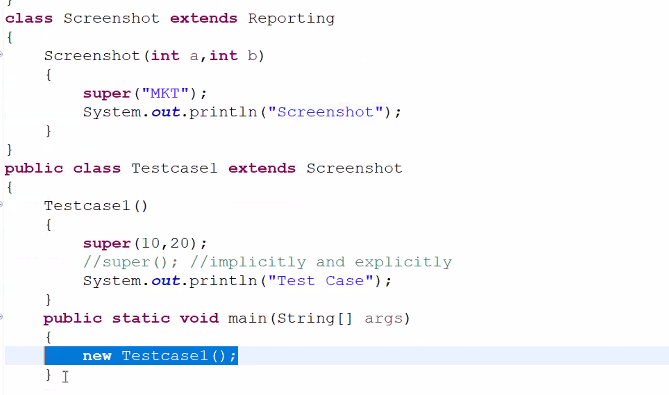
09/25/2024

Type of Super Calling Statement

1. Parameterized Super Calling Statement
   1. Can be written only Explicitly
2. Non-Parameterized Super Calling Statement
   1. Can be written both implicitly and explicitly

It is because the constructor has two types

1. Parameterized
2. Non-Parameterized



What is This calling statement:

Can only be written in Explicit manner

Will be applicable for Single class

If we are having 1k Constructor, then no need to create 1k Object

By using this statement, we can create one Object for all the 1K Constructor.

This calling statement will be the 1st line of each constructor

Any constructor should never be the super and this calling statement

Can we call this

Someone constructor -> Super

Own Constructor -> this

Types of This calling statement:

1. Parameterized
   1. Will call the constructor which is parameterized
2. Non-Parameterized
   1. Will call the constructor which is Non-parameterized