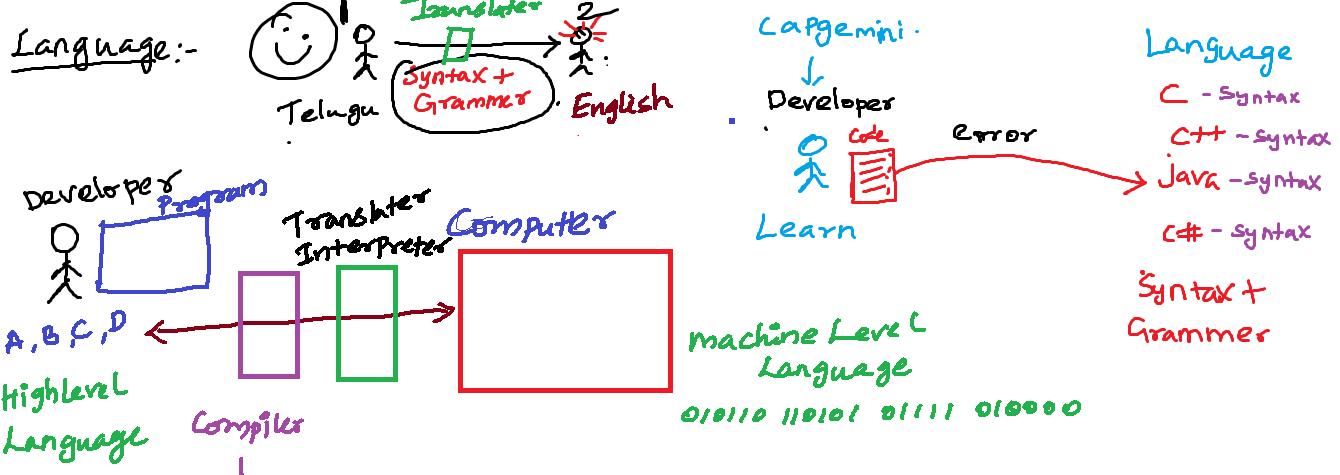
Basics of Language:-



Eclipse IDE:-

Workspace:

It is a folder contains one or more related java projects.

Project:-

It is a folder. It contains one more files related to that project

One or more java files

JRE System Library:-

Has all predefined java classes

Src :-

All user defined .java file must be place inside the src folder

Build -> javac filename.java

Run -> java filename

JDK

JRE

JVM

Sample.class 🡪 JRE 🡪 Machine code(output)

Sample.class 🡪 JDK(JRE), JRE 🡪 Machine code(output)

Sample.java🡪 JDK 🡪 Sample.class🡪 JRE🡪 Machine code(output)

.apk(.classes)

|

Phone (JRE)

|

Machine (Output)

Identifiers:-

All words in java. It is called as identifier

Reserved Keyword or User Defined Keyword

package name, class name, method name, variable name

Package Name:

* All letters must be in lower case
* Should not contains any space

Variable Name:-

* A java variable can start contain (a-z A-Z 0-9). Special character \_ $
* Variable name should starts with a character(a-zA-Z \_ $).
* It should not start with a number
* Variable name should be meaning full name
* If the variable is constant variable then variable name must be capital letter
* If variable name is single word all letters must be smaller case
* If variable name contains more than one word then first word all letter smaller second word onwards each word first letter must be capital

Method name:-

* If method name is single word all letters must be smaller case followed by bracket()
* If method name contains more than one word then first word all letter smaller second word onwards each word first letter must be capital then followed by ()

Class names:-

* Class name each word starting letter must be in capital all the remaining letters in lower case

Interface Name:-

* interface name each word starting letter must be in capital all the remaining letters in lower case

class Syntax:-

package <packagename>;

import <packagename>;

accessmodifier class <classname>{

constructor

variable

methods

}

Class in java:-

A class contains related members

Members – Member variable, Member Functions

Class is template or blueprint of an object

Class is a container in which we write related variable and methods (Encapsulation)

Object Oriented concept:-

Class and Objects:-

Why Functions?

Execute some logic or task

1. Function without arg without return value

**public** **void** multiply() {//perform task or logic

**int** a=29;

**int** b=45;

**int** c=a\*b;

System.***out***.println(c);

}

1. Function with arg without return value

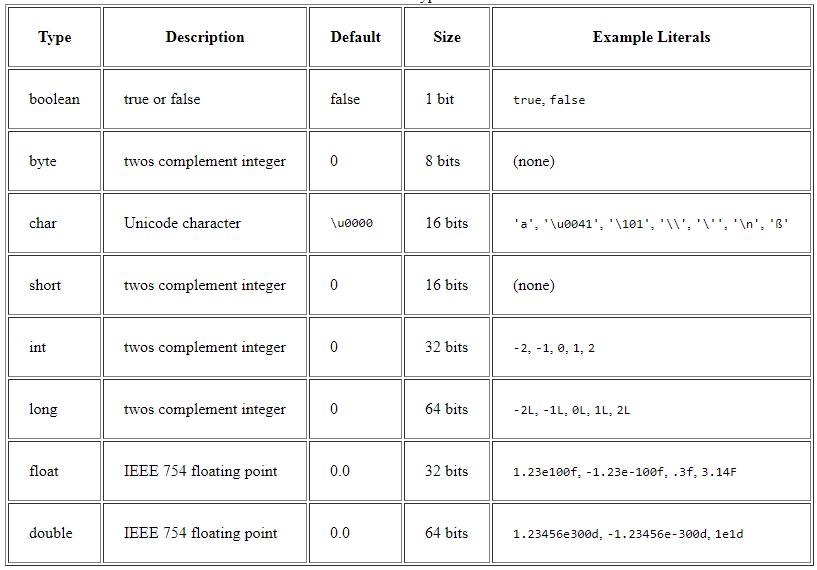
**public void add(int a,int b) { //perform task or logic**

**int c=a+b;// logic**

**System.out.println(c);**

**}**

1. Function without arg with return value
2. Function with arg with return value



Variable in java:-

Variable is an identifier for the memory location

Variable declaration

Datatype variablename;

Variable initialization

Variablename=value;

String name= “rajesh”

String rollno;

String year;

String sem;

int phone;

String email;

float cgpa;

rajesh, 3555SF, IV, VIII, 445454, [rajesh.kit@gmail](mailto:rajesh.kit@gmail), Chennai, 6.4

Types of variables:-

1. Instance variable
   1. Declared inside the class and outside of any method in the class without static keyword
2. Local variable
   1. A variable declared inside the method or block
3. Static variable
   1. Declared inside the class and outside any method with static keyword
4. Reference variable

Types of methods:-

1. Instance method OR non static method
2. Static method OR non instance method

Packages:-

Collections of related java classes, interfaces and enums.

Avoid the naming conflict

Variable declare:-

---------------------------

Accessmodifier Nonaccessmodifier datatype variable;

Class declaration:-

---------------------------

Accessmodifier Nonaccessmodifier class classname{

}

Function:-

Accessmodifier nonaccessmodifier returntype functionName(arg){

-----------------;

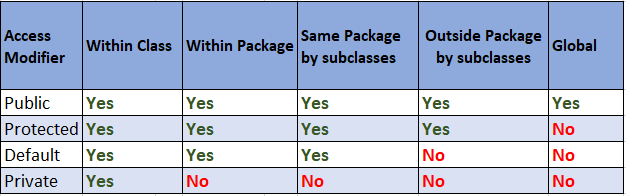
-----------------;

}

Access modifiers:-

private, default, protected, public

* Can be added before variable, method, class , interface



Non-Access Modifiers:-

Non access modifiers changes the default behaviour

Final, static, abstract, synchronized, volatile, native, transient

int mark=100;

Variable initialization:-

* Local variable -> while declaring a local variable we must initialize the value
* Instance or static variable
  + During variable declaration you can initialize

Flow Control

Conditional control statement

If, if else, if elseif, switch

If:-

if(condition){

statements;

}

Un-Conditional Control Statement

Break, label

{

}

Block – grouping set of statements

switch(condition or choice){

case label:

statements;

case label:

statements;

case label:

statements;

default

statements;

}

Java Operators:-

Assignment Operator

Int a+10;

Relational Operator -> comparing between values. It always returns a Boolean true or false

10>20 -> false

10>=20 -> false

10<20 -> true

10<=20 -> true

<,<=,>,>=,==,!= or <>

Arithmetic Operators

+,-,/,\*,%

Conditional Operator:-

(condition) ? value1 : value2; if(condition){

Statements

Else{

statements }

logical operators:

&&, ||, !

Control Statements:-

You can control statement or flow of statement execution

Conditional Statements:- if, if..else, if..elseif..else, switch

Un Conditional Statement:- break, continue, label

Looping statement:-

One or more java statement can be executed finite no of times. You have to loops

For, while , do..while, advanced for loop

For loop:-

for (initialization;condition;increment/decrement){

statements;

}

Advanced for loop:-

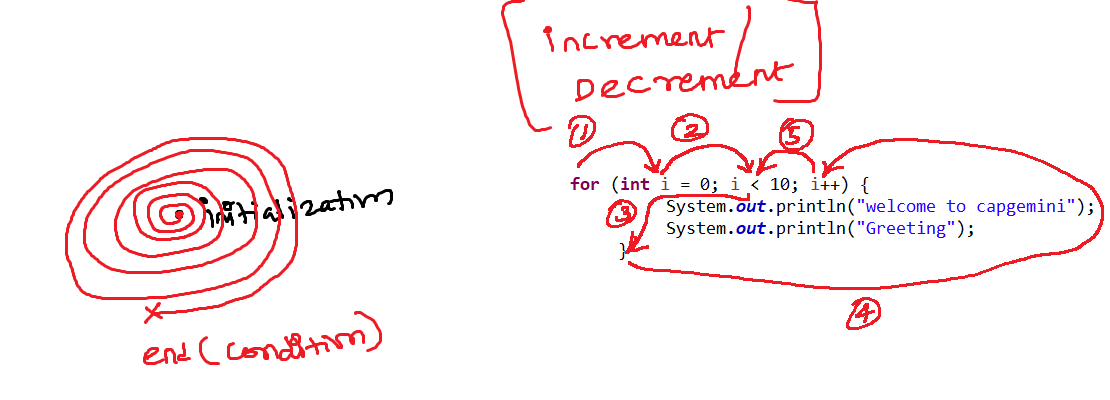
-----------------------------

for(datatype variablename:arrayname){

statements;

}

Instance of Operator:-



while loop:-

while(condition){

statements;

}

do while:-

initiliaztion

do{

statements;

increment/decrement;

}while(condition);

break and continue

Unlabelled break and Unlabelled continue

labeled break and labeled continue

Array:-

Group of similar data stored in a single variable is called as array

Array index starts with zero end with size-1

Eg:-

int[] mark=new int[] {46,34,67,23,78,89};// primitive array

String[] city=new String[] {"","","","",""};// reference type array - String predefined class

Employee[] employees=new Employee[] {e1,e2,e3,e4,e5,e6};//reference type array - Employee userdefined class

Object Programming:-

Classes:-

Encapsulation – data (variable) and the code (method) together kept inside the class it is called encapsulation

Putting all variable and the method inside the class

To make perfect encapsulation

All variable you need to keep it as private

Create a public getter and setter method through which other programs can

Access your data

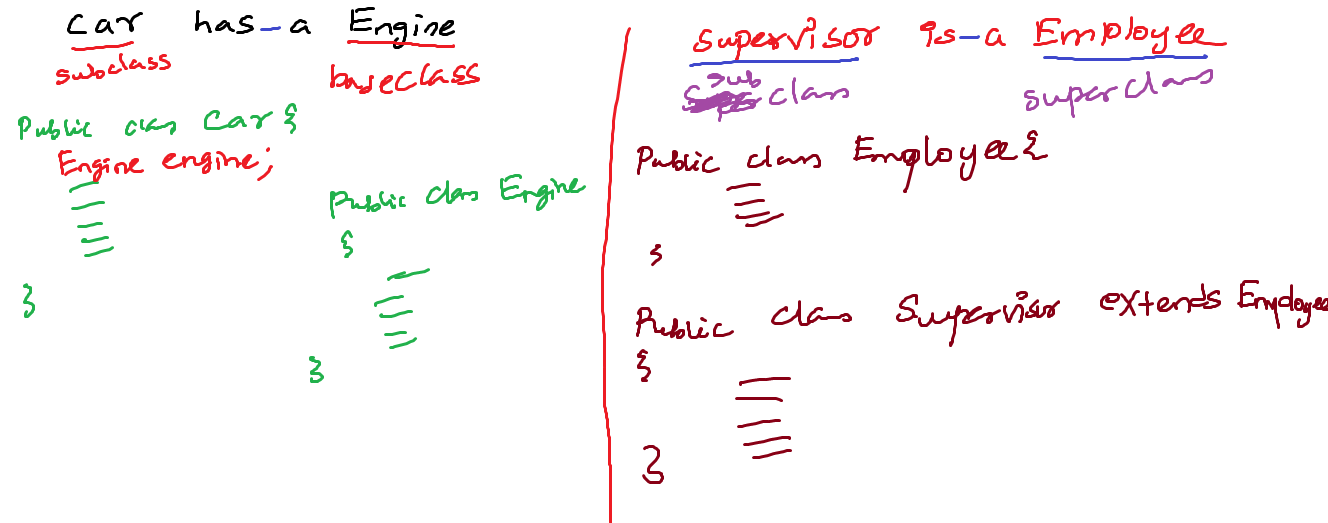
Inheritance can be achieved in two ways:-

Is-a inheritance

If you use **extends or implements** keyword to do a inheritance

Has-a inheritance

If you create an **reference** of one class into another class



Polymorphism:-

More than one form

A task can done more than one form

Send

Person A -----------------------------------------------------------🡪 Person B

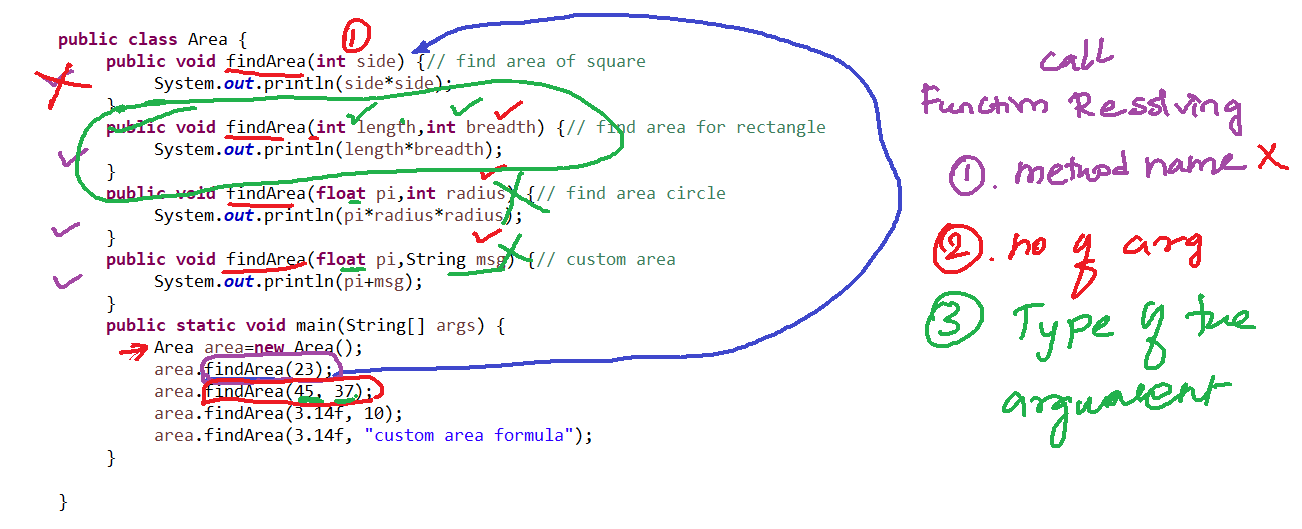
SendParcel (courier)

SendParcel (speedpost)

SendParcel (friends)

SendParcel (myself)

findArea:-



Function call resolving or method binding:-

For the function, call which method implementation or definition get executed decision-making is called function call resolving

If any conflict comes in the function call resolving

The function call resolving decision will be made based

1. Method name
2. No of arguments
3. Type of the arg

Function with same different argument in same or different class it is called as method overloading

Function with same same argument in different class it is called as method overriding

Abstraction:-

Hiding the implementation and showing the essential information is called abstraction

Car has very complex many mechanical parts, as you need to know how to drive the car

Ex:1

**public** **void** findArea(**float** pi,**int** radius) { System.***out***.println(pi\*radius\*radius);

}

This Implementation show all the details it is not abstraction

If method does have any implementation, it is called concrete method or implemented method or non-abstract method

Ex:2

**public** **void** findArea(**float** pi,**int** radius);

It is not the complete implementation and it is showing only the essential information (abstraction)

If method does not have any implementation, it has only the declaration. Abstract method or unimplemented method

Two ways you can implement abstraction:-

1. Abstract class - 100% abstraction
   1. Abstract class does not allow you to create object but you can inherit the abstract class

To other classes

1. Interface – 100% abstraction

If all the methods abstract method

For the abstract class and interface, you cannot create an object. But you can create reference variable

|  |  |
| --- | --- |
| Abstract Class | Interface |
| 1. Abstract class can have concrete or implemented methods | Interface can have abstract method or unimplemented methods |
| 1. An abstract class should have minimum one abstract methods may have zero or more concrete method | All the methods in the interface are abstract methods |
| 1. Inherit the abstract class to other child class **extends** keyword | Inherit the interface to child class **implements** the interface |
| 1. Abstract class methods need to explicitly add abstract keyword for abstract methods | All the method in the interface by default public abstract methods |
| 1. All the variable access modifier non access modifier default set by the developer | All the variable declared in the interface by default public static final variable |