Class and object :

Class is an entity where we can define all the properties in the form of a method and variables

e.g:create a car class

**public** **class** Car {

String model;

**int** year;

**public** **static** **void** main(String[] args) {

//model and year variables are created globally and all the car objects can access these

//object creation for car class

//new car() is the oject

//a,b,c object reference variables

Car a=**new** Car();

Car b=**new** Car();

Car c=**new** Car();

a.model="toyota";

a.year=2015;

b.year=2016;

System.***out***.println(a.year);

}

}

Methods or functions:

1.Main method:starting point of execution

2.Non-static methods:

i.no i/p,no o/p

ii.no i/p,some o/p

iii.i/p with o/p

note:

void-if method does not return anything,void shd be used ---e.g public void add()

retuen type:if it returns some value,the return type of o/p value shd be used

e.g public int add()

how to access non static methods of a particular class?

1.first,chks for main method

2.we shd create an object for the class.

3.now,we can access all the methods by using object ref name.

Syntax:Car a=**new** Car();

a.methodname();

//a-obj ref name

e.g:

**public** **static** **void** main(String[] args) {

methods m1=**new** methods();

**int** x=m1.add(10, 20);

System.***out***.print(x);}

**public** **int** add(**int** a,**int** b)

{

**int** c=a+b;

**return** c;

}

//since non static method,we need to access the method using obj reference

Global variable vs local variable

Global: also known as class variable

Declared immediately after class and outside main method.

It has scope throughout the class

Local: inside a particular method either main method or user defined method

How can we access global variable in main method?

By creating object reference

Public class{

Int I;//global var

**public** **static** **void** main(String[] args) {

Class c1=new class();

System.***out***.print(c1.i);//accessing global var thru obj ref

Method overloading:

Note:1. methods are always independent,we cannot create a method within a method

2.we cannot create duplicate methods

But we can create duplicate methods using method overloading.

What is method overloading?

Different methods with same names with diff parameter type or with different no of arguments.either no parameter,one i/p parameter,2 i/p parameter

There shd not be two methods with same no.of parameters

If there is same no of parameters,then parameter type shd br diff

**public** **static** **void** main(String[] args) {

Methodoverloading obj=**new** Methodoverloading();

obj.sum();

obj.sum(2);

obj.sum(3, 2);

}

**public** **void** sum()

{

System.***out***.print("sum");

}

**public** **void** sum(**int** i)

{

System.***out***.print("sum="+i);

}

**public** **void** sum(**int** x,**int** y)

{

System.***out***.print("sum="+(x+y));

}

Static and non-static methods:

How to call non-static methods?

1.create object for class and access methods with obj reference variable

How to call static methods?

1.direct calling: just the method name

2.call by class name :classname.methodname

Note:static variables and static methods are accessible by objects ,but not a good practice to access thru objects.best practice is to access it using direct calling(within same class) or call by classname.methodname(for parent-child class concept)

Wrapper class :

Wrapper classes are used to convert from one data type to another .

i.e from string to int,string to Boolean,string to double, int to string etc

String to int

String x="100";

**int** y=Integer.*parseInt*(x); //parseint func is used to convert string to int and it is available in Integer wrapper class

System.***out***.println(y+20);

String to double:

**double** y=double.parsedouble (x);

Int to string:

**int** a=100;

String z=String.*valueOf*(a);// value of is used to convert int to string and it is available in String wrapper class

System.***out***.println(z+20);

Summary:

Lacal variable vs global variable

1.if a variable is declared locally,within main methos or someother method,it can be accessed directly.

String *name*="Honda";

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

2.If a variable is declared globally,then it can be accessed in two different ways:

1. Static keyword and declared globally:

Static String *name*="Honda";

In this case,it can be accessed without any object creation .accessed directly with variable name.

**static** String *name*="Honda";

**public** **static** **void** main(String[] args) {

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

}

ii)without static keyword and declared globally:

Then,we need to create object for d class and access them using objects.

**public** **class** Car {

String *name*="Honda";

**public** **static** **void** main(String[] args) {

Car c1=new Car();

System.***out***.println(c1.*name*);

}

}

Consider this program:

1) **public** **class** Car {

**static** String *name*="Honda";

**public** **static** **void** main(String[] args) {

*name*="toyota";

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

}

}

o/p: Toyota

2) **public** **class** Car {

**static** final String *name*="Honda";

**public** **static** **void** main(String[] args) {

*name*="toyota";

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

}

}

o/p: error will be thrown saying “The final field Car.name cannot be assigned”,

because once declared final,its value cannot be reassigned.