OBJECT:

RENT a ROOM

Public class room

{

create the object for electricity : U can use electricity :Data members and function members

create the object for Water: U can use water :Data members and function members

}

class electricity {}

class water {}

class DTH {}

What is object ?

-The object is the leap memory location. to initialize "Function members", "Data members" of same class or different class.

Why to create the object ?

-To obtain the objected oriented structure. like said in above example, by creating the object User can use :Data members and function members of same class or different class.

When to create the object ?

-When user want to use the "function member" or data members of same class or different class

What we access by creating the object ?

-By creating the object we can access the "NON STATIC" function members, "NON STATIC" data members of same class, and "NON STATIC" function members, "NON STATIC" data members of different class

**Only CLASS VARIABLES can be Accessed in Other class. by creating object.**

CONSTRUCTOR:

public class sample

{

int age;

age=10; // this is wrong

PSVM(String args[]) { } }

What is constructor ?

-Constructor is used to construct the value. In the above example variable age cannot immediately signed/initialized with value.

sample()

{

} constructor "sample" must be used which is default/implicit constructor.

Types of constructor ?

-Default/implicit

Parametrized

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

What is the use of Parametrized constructor ? in real time ?

class\_name ref\_variable=new class\_name(args)

-in the argument section if i pass the values it will directly invoke the constructor only

-If u ask me if there is two constructor with same arguments data type : X it is not possible.

e.g sample(int a){}

sample(int z) {} :X not allowed

-If the CLASS has taken step to create parametrized constructor. Then USer can create ONLY that much of OBJect creation as much parametrized constructor is Created

e.g sample(int a){}

sample s1=new sample(10); : Only this is allowed

sample s2=new sample(1000000); : Only this is allowed ; so called constuctor overloading

sample s2=new sample(); X not allowed as there is no constructor for this.

**IMP: const don't need again object or obj ref variable to initialize or override the value.**

**Const means constructing values**

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

STATIC AND NON-STATIC:

What is static variable ?

-Static variable means fixed memory location, AND it retain its previous value UNtil U or constructor override it

e;g static count i=10;

sample() //constructor

{count++;

Sop(count)}

PSVM{ //main method

sample s1=new sample(); //count value increase to "11" and print is also "11"

sample s2= new sample() //count value increase to "12" and print is also "12"

sample.count=20;

What Is NON Static variable ?

-NON-static variable means dynamic memory location, AND it DOES NOT retain its previous value

e;g count i; // As it is non-static no-intailization // infact it doenot have memory

sample() //constructor

{count++;

Sop(count)}

PSVM{ //main method

sample s1=new sample(); //count value inc to "1" and print is also "1" //doenot have memory

sample s2= new sample() //count value inc to "1" and print is also "1" //doenot have memory

sample.count=20

// In the non static every time object is created new memory location.

NON-STATIC method access features ? in same class

-direcly call/declare/override both NON-static or static method/variable

STATIC method access features ? in same class

-direcly call/declare/override static method/variable

But in STATIC methods object should be created to call/declare/override non-static method/variable

THIS KEYWORD:

It is better approach to use meaningful names for variables. So we use same name for instance variables and parameters in real time, and always use this keyword.

What is "this" keyword?

- "this" keyword is used as instance of the present class

note: this keyword by default present while calling non-static method/variable in non-static method.

What is the use of "this" keyword ?

-By the use of "this" keyword the class constructor and class variables/global variables can have same variable names.

e.g class A

int a;

A(int a){this.a=a;}

PSVM()

{A a1=new A(100);

}

What is "this()" method ?

-"this()" should be first statement.

If this() IS called in the current constructor, then it is only interested in calling the other constructor, with or without argument.

A(double c){}

A(String a, int b ){this(double)}

What is the use of "this" method() ?

**-this() means calling constructor only.**

**control statements:**

**If statements**

**if(condition) {}**

-code will enter the loop if condition is met

**if else statement**

**if(condition) {}**

**else{}**

-if the condition is false code will jump to else loop.

**if else if statement**

**if(condition) {}**

**else if(condition) {}**

**else {}**

-if the 1st condition is false code will jump to 2nd else if condition go on until condition is met if not else statement is met.

**nested if statement**

**if(condition)**

**{**

**if(condition)**

**}**

-condition inside a condition is nested if

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

**for loop**

for(int g; condition i;e < or> iteration number; decr/incr )

- for loop is used for fixed iteration

**Enhanced for loop:**

for(int g : expression) // here expression is array variable.

-enhanced for loop is used for arrays

e.g

int a []={10, 20, 30,40}

now to iterate a[] from for loop

for (int b=a)

{sysout(b)}

**labelled for loop:**

a:

for(int g; condition i;e < or> iteration number; decr/incr)

{

b:

for(int g; condition i;e < or> iteration number; decr/incr)

{

break a;

}

}

-here for loop is inside another for loop. if inner for loop break it will fall into outer for loop.

to exist from outer for loop need to break by using label name.

**while:**

while(condition)

- if the iteration is not fixed go for while loop.

while(true)

**do-while:**

do{

}while()

- if the iteration is not fixed but need to execute code atleast once go for do-while

**Switch:**

switch(expression/variable)

{

case expression\_value :

break;

}

- to switch to expression declared with different cases.