* Setter and Getter Method
* Constructor
  + Default
  + Parameterized

Class use static {

Static int a=3;

Static int b;

Static int meth(int x)

Sopln(“Val of x”+ x);

Sopln(“Val of b”+ b);

Sopln(“Val of a”+ a);

Static{

Sopln(“Static block initialized”);

}

PSVM (){

Meth(42);

}

In call by value only copy of args is used and what occurs in para that receives the args has no effect outside

* == tests if both references refer to same object in memory And s1.equals(s2) doesn’t.
* StringBuffer class is mutable i.e. StringBuffer Capacity can be changed.
* Create an employee exception class whose constructor receives a string that consists of emp\_id and pay\_rate. Save the file as employeeException.java create an employee class with 2 fields id\_num and hourly wage. The employee constructor requires values for both fields. Upon construction throw an employee exception if the hourly wage is less than 6$ or over 50$. Save the class as employee.java

Write a program that establishes at least 3 employees with hourly wages that are above below and within the allowed range. Display an appropriate message when an employ is successfully created and one is not. Save the file as throwEmployee.java

In case of final

Scope of final is limited to each object separately.

If a final is defined in class A and class B extents from class A, u can’t change that final value.

Static is bound to class and instance is bound to object.

In overriding if return type is changed then it’s called covariant return type.

**Abstract Methods**

A method with no body.

It’s declared with abstract keyword.

U cannot instantiate an object for an abstract class.