STATIC BLOCK.

Requirement

```
Class Temp
{
static int x=10;
}
```

```
Class Temp1
{
psvm( )
{
Sop(Temp.x);
}
}
```

```
Class Temp2
{
psvm ( )
{
Sop(Temp.x)
}
}
```

★ Static Initialisation of Static datamembers
→ as it has same value always.

For Dynamic Initialisation of Static Data Mem.
↳ Have different - different values every-time

```
class Temp
{
Static int x;

Static
{
try
{
x = System.in.read();
}
Catch(Exception e)
{
}
}
}
```

```
Class Temp1
{
psvm()
{
Sop(Temp.x);
Sop(Temp.x);
}
}
```

Both Sops print same values, as class is ~~only~~ loaded only once.

```
Class Temp2
{
psvm()
{
Sop(Temp.x);
}
}
```

- Static Block is used to initialise ~~the~~ the Static Data Members Dynamically.
- Static Block is always executed at class loading time. That means before the execution of a main f^n.
- Static block is always executed only once in the life cycle of a class.
- If you want to perform any task only once in the life-cycle of the class, then put the code of that task

in a static block.

IQ → ~~I~~ Can you executea java ~~do~~ class without a main f^n?

Ans → No, not now ~~after~~ from JDK 1.7
It was possible until JDK 1.6 because of bug.

- No class can be executed in Java without a main f^n from JDK 1.7 onwards.

①
```
class StaticC
{
static int x;
static
{
Sop("static block");
try
{
x = System.in.read();
}
catch (Exception e)
{

}
}
}
```

```
Class StaticTest1
 {
~~Publi~~ psvm ('  ')
  {
  Sop (StaticC.x);
  }
 }
{e
Class StaticTest2
{
psvm (     )
{
Sop (StaticC.x )
 }
 }
```

②

```
class Temp
{
static Demo d;
static
{
d = new Demo();
}
}
```

```
Class Demo
{
Void show (int x)
{
Sop(x);
 }
 }
```

```
class Temp1
 {
psvm()
 }
Temp.d.show
 (10);
Sytem.Out
.Println(10);
 }
```

```
Class System
{
static PrintStream Out;
 static
 {
 Out = new PrintStream();
  }
```

```
Class Printstrean
 {
publicVoid
println (int x)
 {
 =
 =
 }
```

Que→ How 'System.Out.println' combination is made?

Ans:→ System → It is a Class.

Out → It is a static Data Mem. of 'System' class. & Ref. Variable of 'PrintStream' class.

Println → It is a f^n of 'PrintStream' cless.