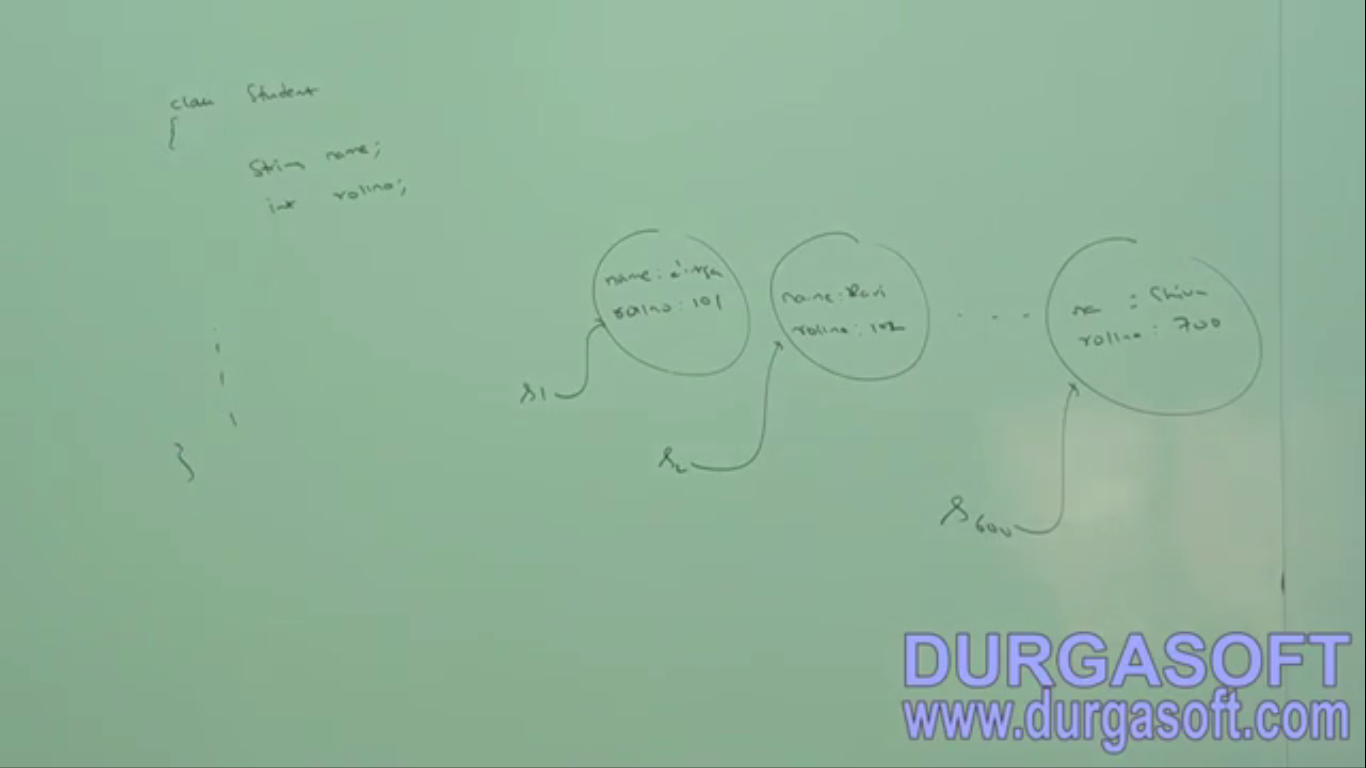
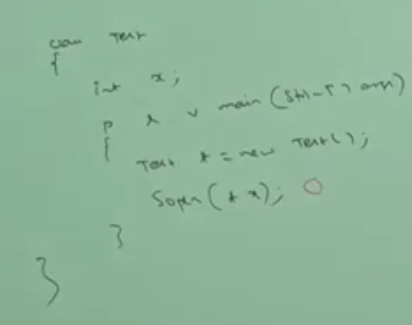
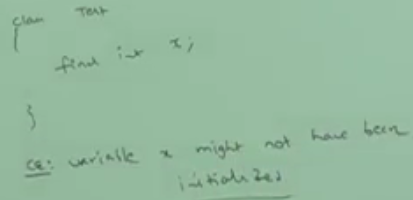
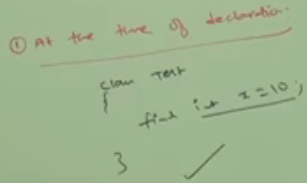
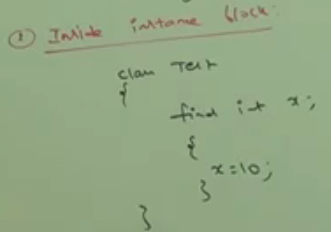
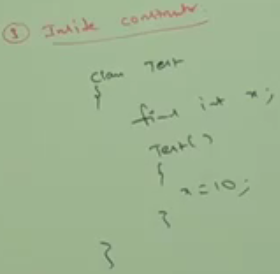
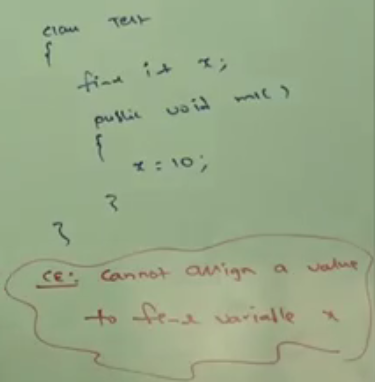
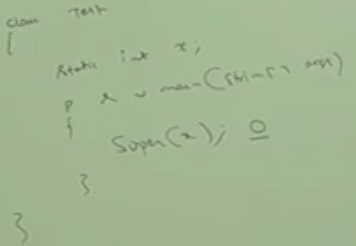
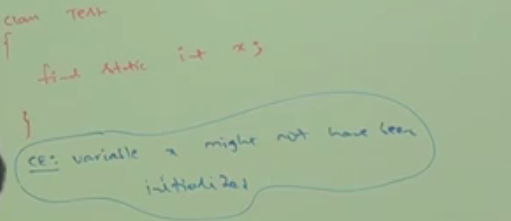
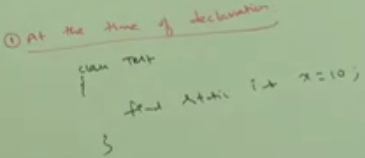
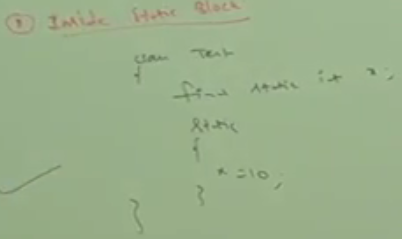
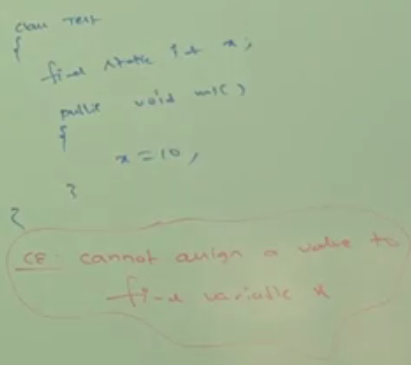
final variables

1. 3 kinds of variables
   1. instance variable
   2. static variable
   3. local variable
2. Now attaching final keyword with each of 3 kinds of variables

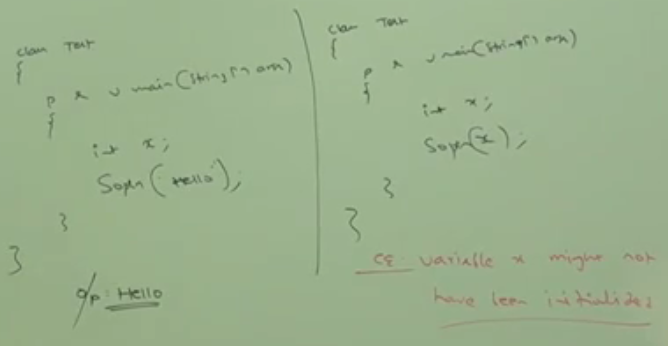
final instance variable

1. **Instance Variable**: If the value of a variable varies from object to object such type of variable is called **instance variable**.   
   For every object, a separate copy of instance variable is created.   
   
2. For instance variable, we are not required to perform initialization explicitly. JVM will always provide default value.   
   
3. **final instance variable**: If the instance variable has been declared as final, then compulsory we have to perform initialization explicitly whether we are using or not and JVM doesn’t provide default as in case of non-final instance variable.  
   
4. **RULE:** For final instance variable, compulsory we should perform initialization before **constructor completion  
   When what are various places?**
   1. At the time of declaration.  
      
   2. In instance block  
      
   3. Inside Constructor  
      
5. Is method appropriate place for final instance variable initialization?  
   
6. d

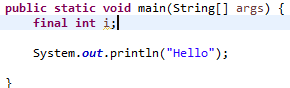
static variable

1. **Definition**: If the value of a variable is not varied from object to object such type of variable are not recommended to declare as instance variable.   
   We have to declare those variables as class level variable as **static modifier**.
2. **Difference b/w instance and static variable.**
   1. In the case of instance variable, for every object a separate copy of instance variable is created.
   2. In the case of static variable, a single copy of static variable is created at class level and shared by every object of that class.
3. **initialization**: for static variable, it’s not required to perform initialization explicitly. JVM will always provide **default values**.  
   
4. **What if static variable is also final?**
   1. JVM will not provide default value. We need to initialize explicitly.  
      ****
5. **Rule**: for final static variable, compulsory we should perform initialization before class loading completion that is the following places are for this.
   1. At the time of declaration:  
      
   2. Inside static block  
      ****
6. **In method, final static variable initialization is not allowed  
   **
7. **d**

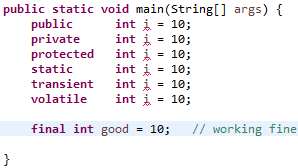
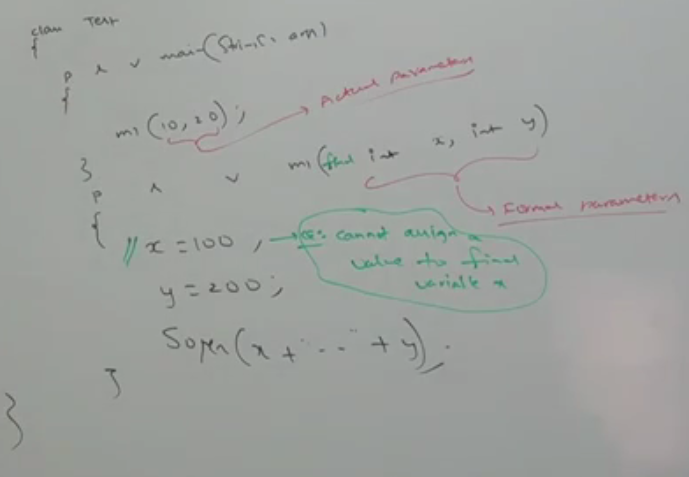
Local Variable 🡪

1. **Definition**: Sometimes, to meet the temporary requirement of a program, we have to declare a variable in the following places such variables are called local variables or temporary variables or stack variables or automatic variables.
2. **Local variables places**
   1. method
   2. constructor
   3. static block
   4. inside loop 🡪 for(**int i; …;…)**
3. **Local variables also called**
   1. Temporary Variable
   2. Stack Variable
   3. Automatic Variable.
4. **NOTE**: For local variable, JVM will not provide any default values. We are responsible to initialize local variable compulsorily before using local variable. If we are not using then it’s not required to perform initialization for local variable.  
   
5. **f**

final local variable

1. Even though, local variable is final, before using only, we have to perform initialization.   
   

NOTE: If we are not using then it’s not required to initialize final local variable. ☺

1. The only applicable modifier for local variable is **“final”**. If we try to apply any other modifier then we will get compile time error.   
     
   NOTE: We were not declaring any modifier then by default it’s default but this rule is applicable only for instance and static variables/ methods.
2. **Formal Parameters**: considered to be local variables of that method. Hence those can be declared as **final**. If formal parameters are final, then within the method, we can’t perform reassignment.   
   ****