**What is generic (came in 1.5 version) and it’s importance**

**Answer: if we specify**

**ArrayList list= new ArrayList();**

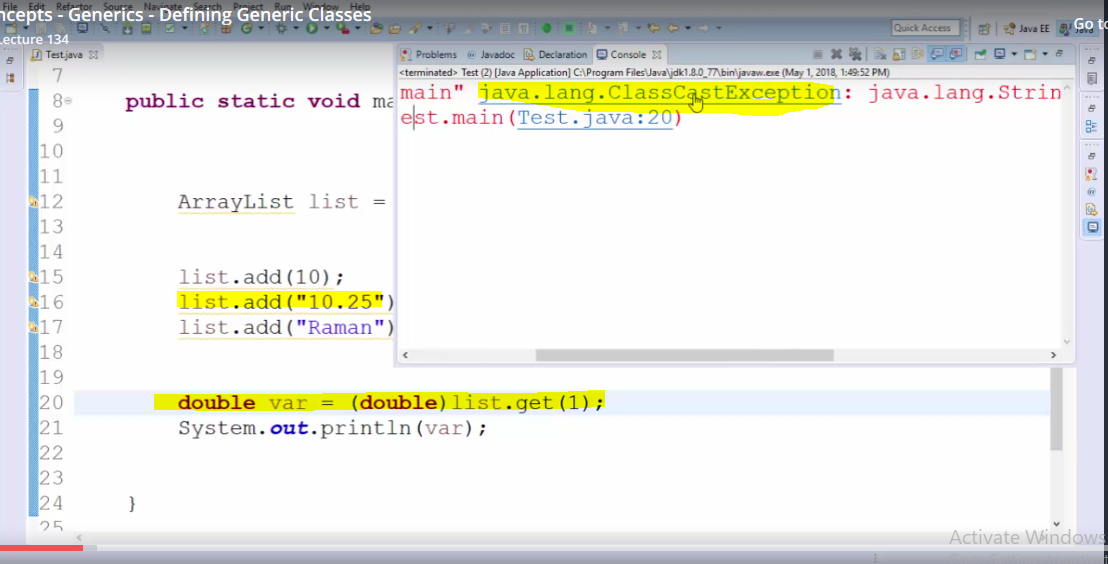
**Here, we are not specific which data type it will be taking**

**Now if we specify,**

**ArrayList<String> list= new ArrayList<String>(); Here, we are saying that array list should take data type of ‘String’. This is one form of generics.**

**Question: What sort of issues we face if we don’t specify generics**

**Answer: If we don’t use generic then we may face type safety issue. Look at the following example. Here we are trying to type cast the “Object of String” into double primitive type. Which is not** allowed due to type safety. Only object of “double” can only be type casted to double primitive type.

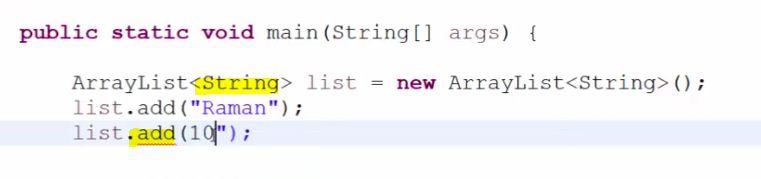


**Due to ‘Type safety’ issue, we use generics. Because, at compile time, it is not known type of value is stored.**

**Generic define**

**ArrayList<String> list= new ArrayList<String>();**

**Now here list is going to accept only string not any other type. If we try to add any other type other than string then it is caught in compile time. This is the benefit of generics. Look at the following example:**



**Another important benefit is that when above generic is defined, then while getting its value then we don’t need to type cast it and it will resolve the type casting issue as we are forced to add the values respective of its class type only. Earlier, when the list without generic was defined then we have to type cast it’s each and every value (whether it is integer, double or string) with its respective object/class.**

**For example: Non generic list**

**ArrayList list1= new ArrayList();**

**List1.add(1);**

**List1.add(“Hitesh”);**

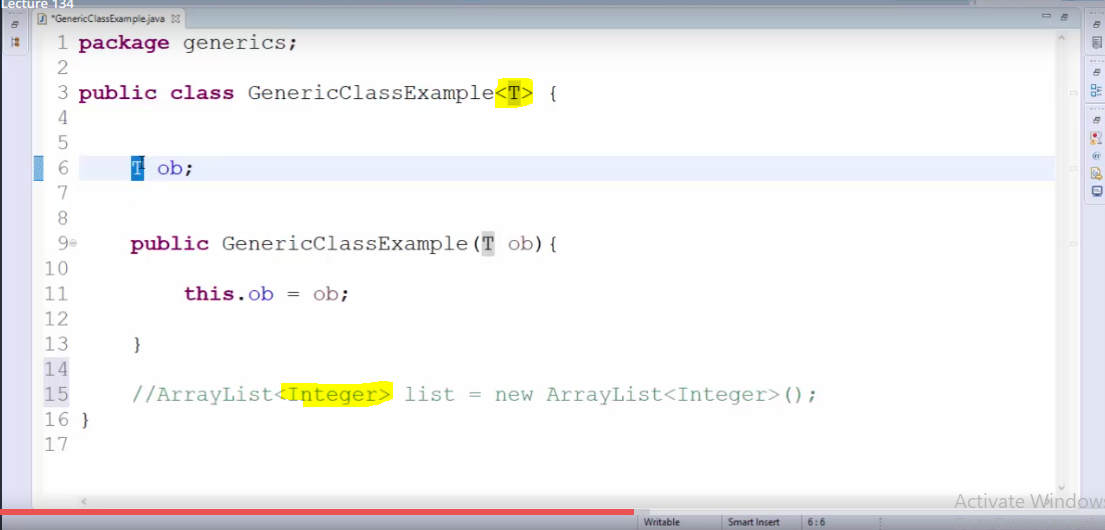
**List1.add(“10.55”)**

**Now, if we have to get first index value which is of type of integer then we have to type cast it as follow**

**int i= (Integer).List1.get(0) and similarly for String and double values as follow:**

**String s= (String).List1.get (1) etc. etc.**

**Generic Class example:**



**Generic methods:**

