

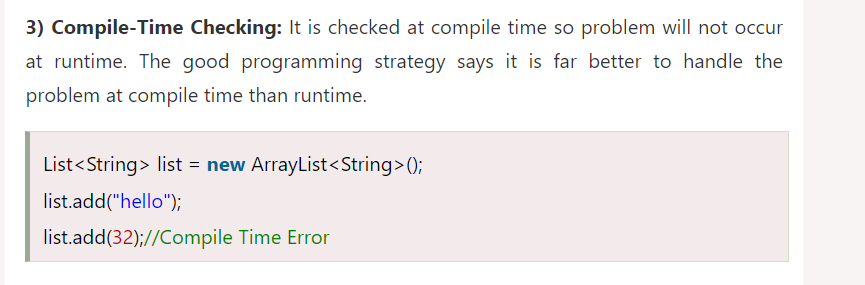
Here we were able to specify the data type that can be stored in the array list via <Integer>

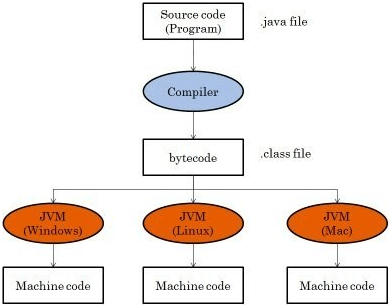
This functionality is generics

Now what is happening internally in above?

During compile time here a reference variable is created in the stack memory names list2, and it is guranted that this reference variable will point to an ArrayList object later on during run time in the heap and that object will contain only the Integers data members

The biggest advantage of using generics



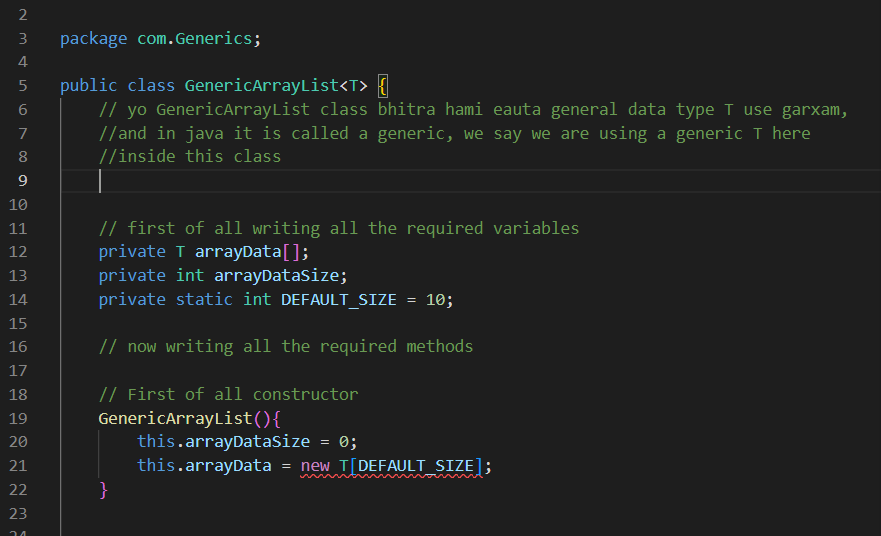


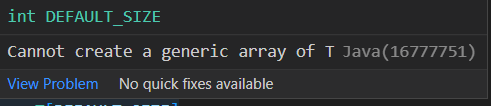
Here the source code after compilation a .class byte code is generated by the compiler

And that byte code is run by the various JVM

Since we know object creation happens at run time

That mean no objects are created in the byte code



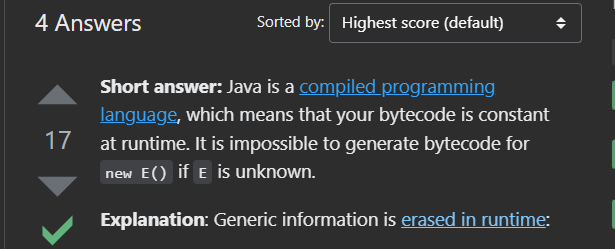


We are getting an error like this

This is because at compile time java lai <T> vanaeko k ho? vannie thaha hunu parxaa, thaha vayaenaa vannae error dinxaa

Here when byte code is generated it does not know what is <T> hence T is unknown when the byte code runs at run time, so it throws error

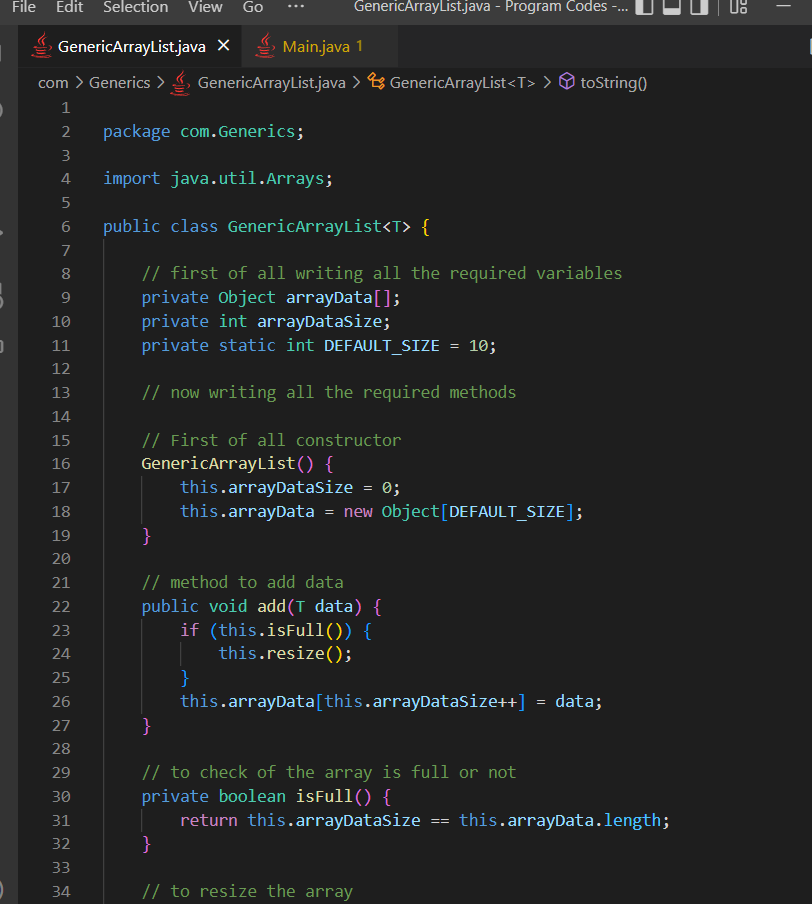
We cannot create an instance of generic type, because the byte code has no idea about T, and T is unknown at run time

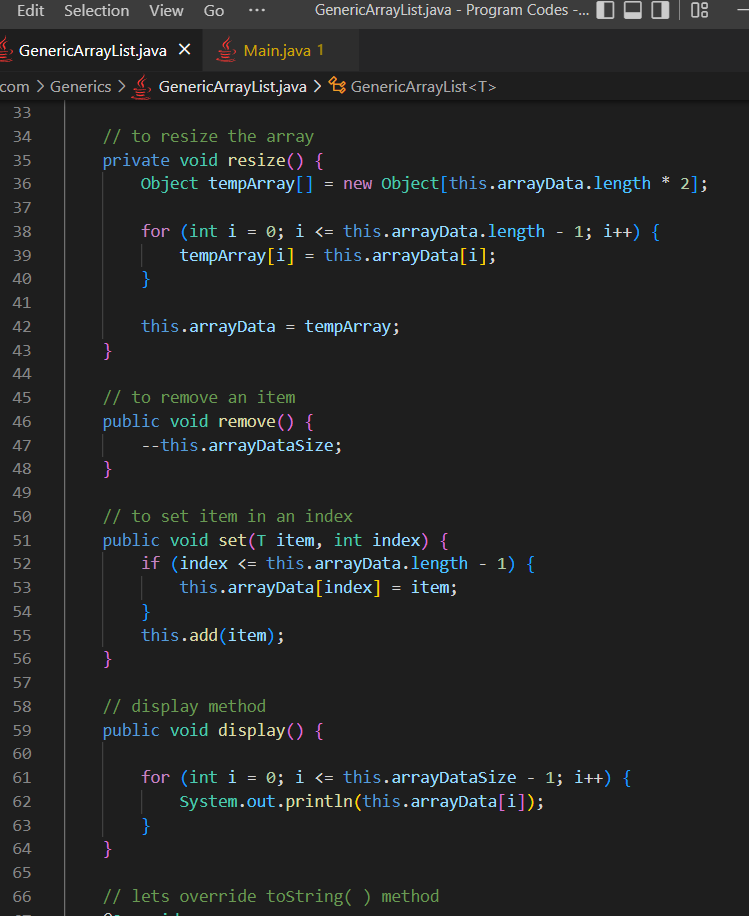


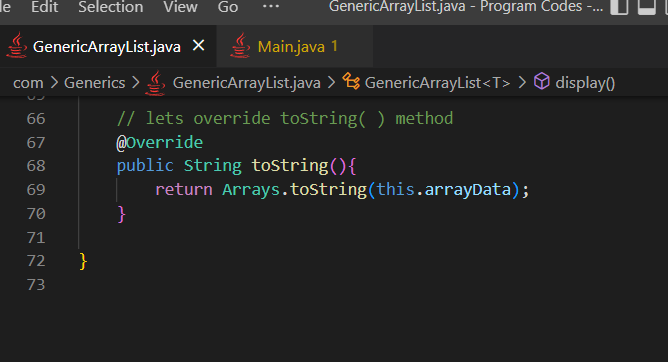
As per this official documentation

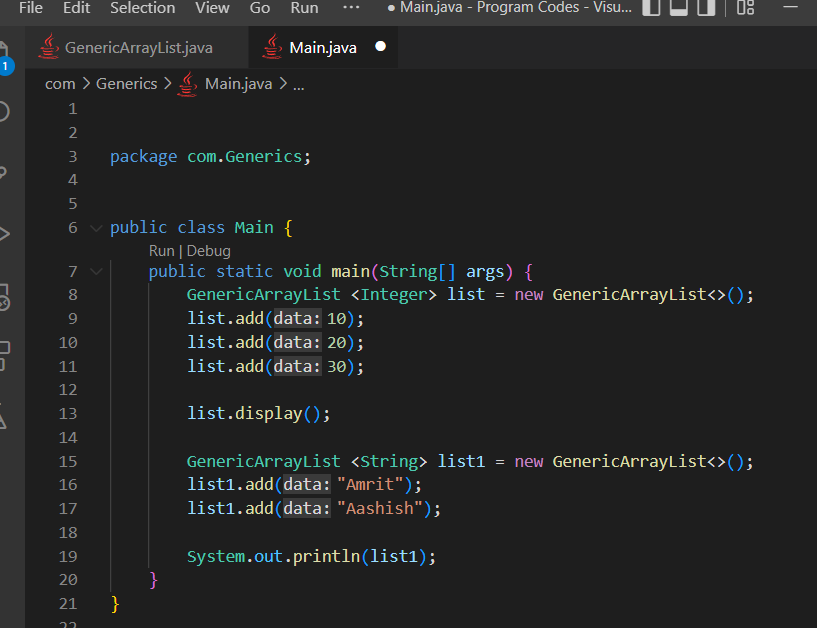
<https://docs.oracle.com/javase/tutorial/java/generics/genTypes.html>

Here use the data type Object instead of <T> to remove above error









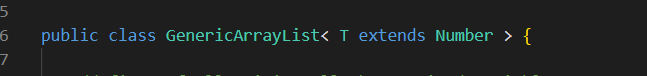
**Providing more restrictions to the <T>**



Here there is no any restrictions for T

T can take any data type during object creation

We can actually restrict what data types are allowed to pass as T



Now T is restricted

Now we can either pass Number or any sub class of Number as data type instead of T

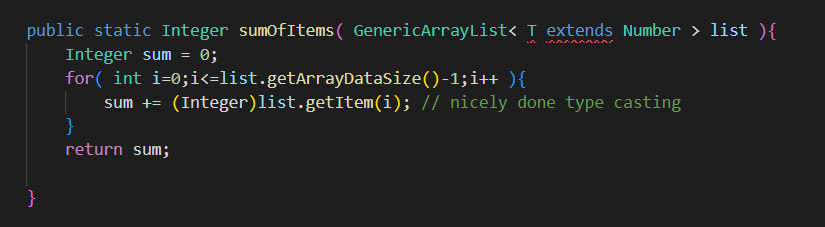
We can not pass something like Strings, it will give us error, but in upper one we could pass anything

**Static methods inside generic class**

The value for the generic data type T is passed during object creation

Static methods do not depend upon objects

So generic T will not work inside the static methods

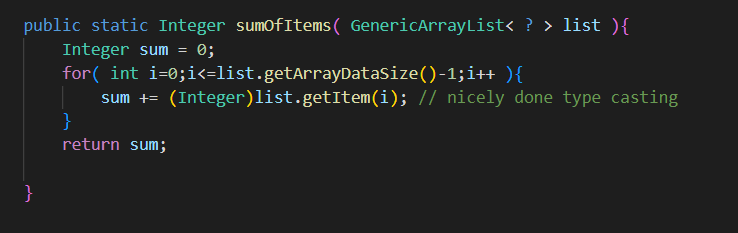


**Upper bounded wildcard, use of extends keyword with ?**

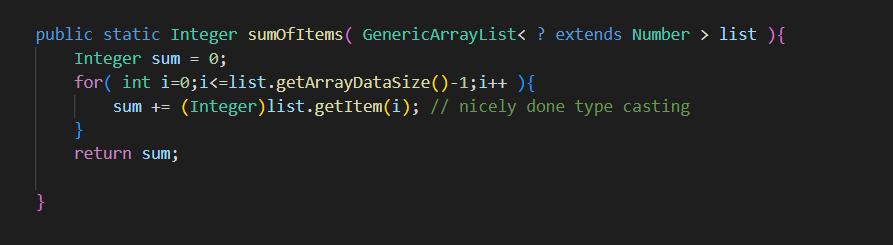
We could use wild card here

? is known as wildcard in java

? actually means an unknown data type



Here sumOfItems( ) method can take list object, but it’s inner data values can be of any data type

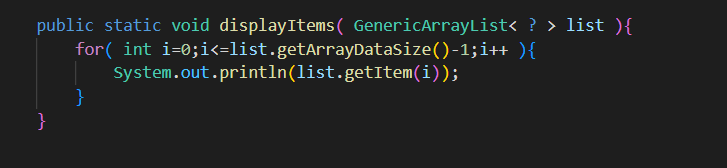


Here < ? extends Number > means :

Now static method will take only those list objects having its internal data values of data type either Number or any sub class of Number

*sumOfItems( ) lae expect garxaa ki testo list object aayos, jasko internal data values either Number data type ko hos or Number class ko sub class type ko hos*

**Unbounded wildcards, use of only “?”**



Here < ? > means :

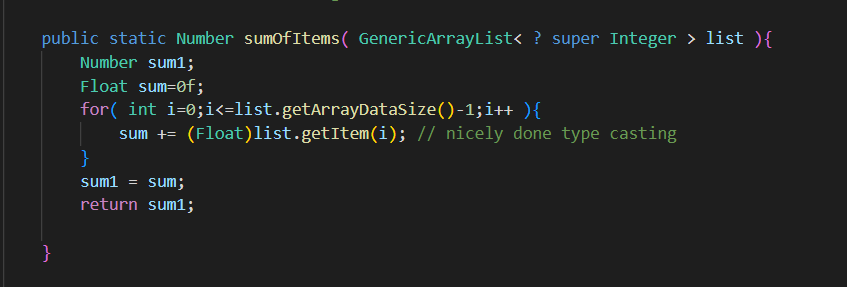
Here the data type of the items inside the list is unknown to the displayItems( ) method

OR

That can be of any data type

*Here displayItems( ) lai thaha xainaa ki list object ko inner data values kun data type ko aauxaa vanaeraa*

**Lower bounded wildcards use of super with?**



Here < ? super Integer > means :

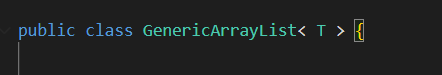
The list object can have inner data values of either Integer type OR the super class of Integer which is Number type

*sumOfItems( ) lae expect garxaa ki testo list object aayos, jasko internal data values either Integer data type ko hos or Integer class ko super class type ko hos*

**Conclusion**

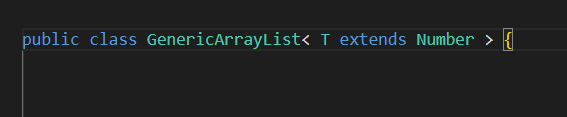
**Unbounded generic**

T can be of any type, no any restriction



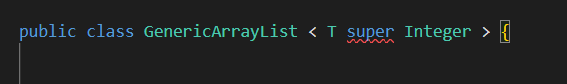
**Upper bounded generic**

T can be of Number class data type or the sub classes of Number



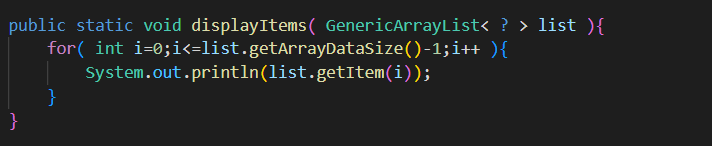
**But in classes we can not do something like this**

**We can have lower bounded wildcard generic in static method**



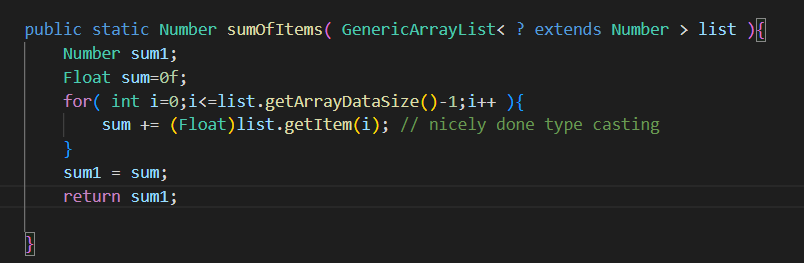
**Unbounded wild card**

The inner data values of list objects can be of any type, the static method has no idea about it



**Upper bounded wildcard**

The inner data members of the list object can be either of Number type or the sub classes of Number i.e Integer, Float, double etc.



**Lower bounded wildcard**

Here inner data members if the list object can be of either Integer type or the super class of Integer which is Number class type

