**National University of Computer & Emerging Sciences, Karachi**



**Computer Science Department Fall 2023, Lab Manual – 12**

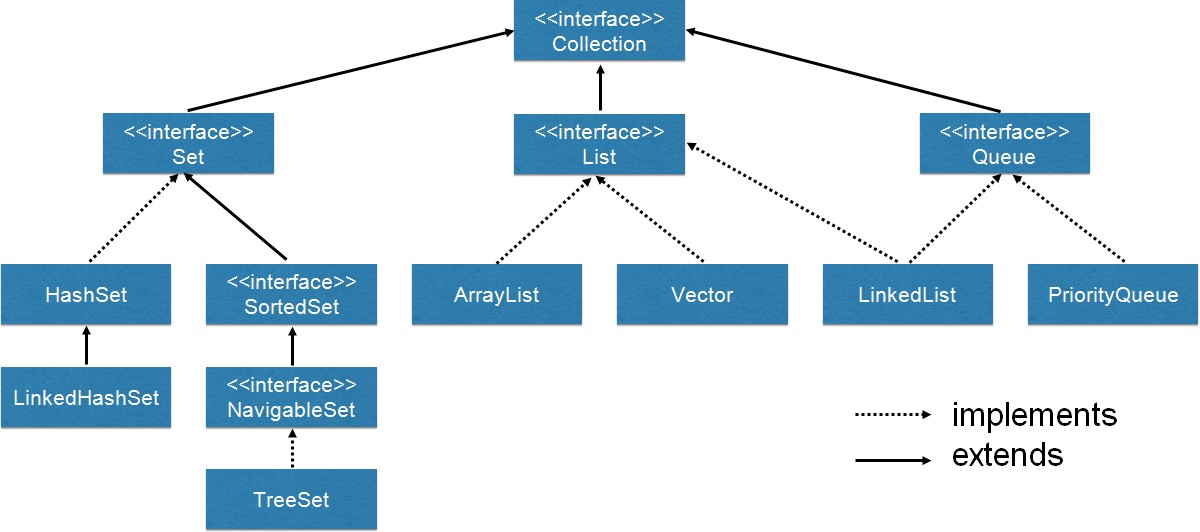
|  |  |
| --- | --- |
| **Course Code: CL-217** | **Course : Object Oriented Programming Lab** |
| **Instructor(s) :** | **Shafique Rehman** |

LAB - 11

Generics in Java

## Collection Framework:

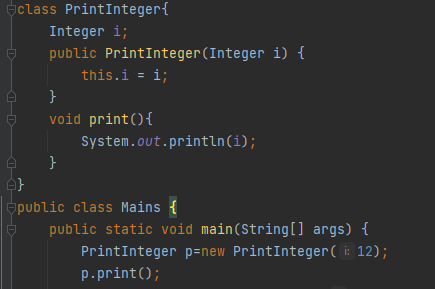
The Java collections framework provides a set of interfaces and classes to implement various data structures and algorithms like Stacks, queues, arrays, linked list etc.



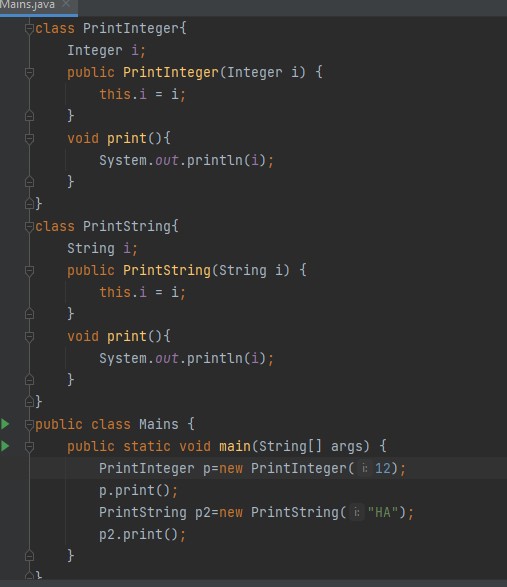
## Generics:

**Generic Classes:**

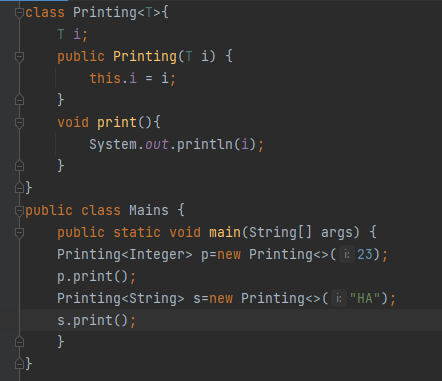
Consider an example, let’s say I want a class that prints an integer value. So, I write a piece of code like this.



Now, for printing a String object we have to replicate the code for String class and after that for other data types as well.



So, it’s better to make use of Generics concept in Java. Here we will make a generic class and use it for printing objects of different data types. The class name is followed by a type parameter section. The type parameter section of a generic class can have one or more type parameters separated by commas.

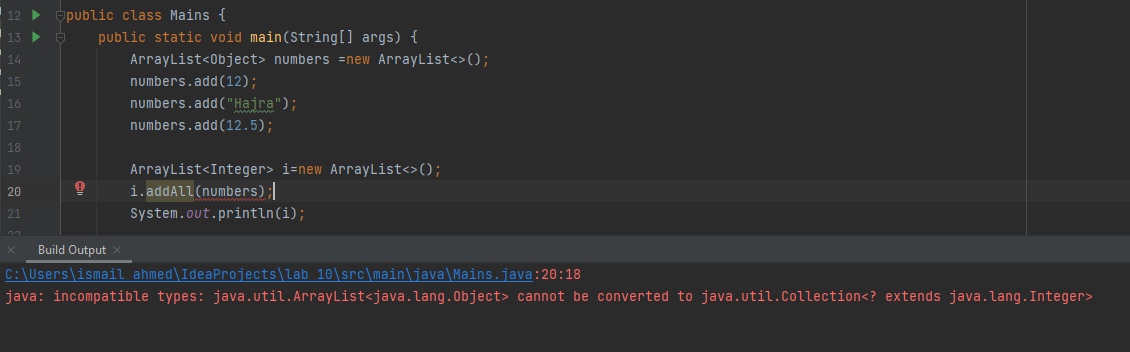


Generics are frequently used with collection frameworks in Java. For example,



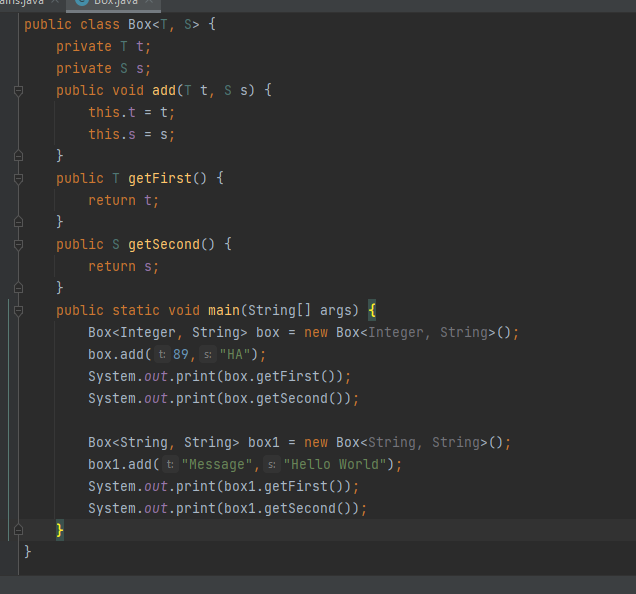
Generics avoid the type Cast Exceptions as they are type safe.

For example, you might think that lets create an array list of Object type and add any type of Item to it. But then type casting error might occur. Shown in example below;



So, generics ensure type safety as well.

A generic class may have multiple parameters.



### Type Parameters

**The common type parameters are as follows:**

T - Type

E - Element K - Key

N - Number V - Value

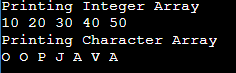
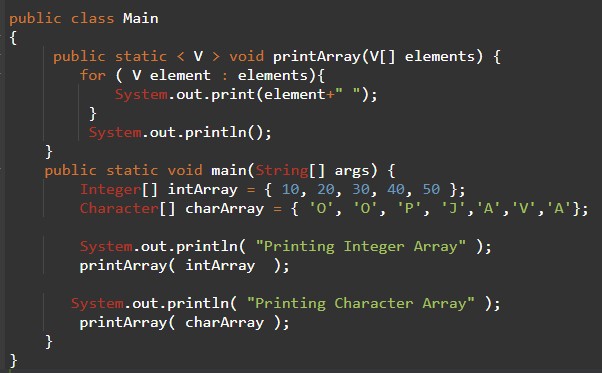
## Generic Methods:

We can also write generic functions that can be called with different types of arguments based on the type of arguments passed to the generic method. The compiler handles each method. Following are the rules to define Generic Methods :

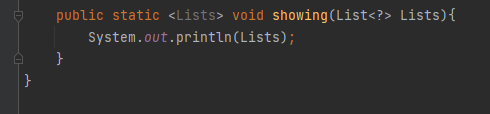
* All generic method declarations have a type parameter section delimited by angle brackets (< and >) that precedes the method's return type ( < T> in the example shown below).
* Each type parameter section contains one or more type parameters separated by commas. A type parameter, also known as a type variable, is an identifier that specifies a generic type name.
* The type parameters can be used to declare the return type and act as placeholders for the types of the arguments passed to the generic method, which are known as actual type arguments.
* Note that type parameters can represent only reference types, not primitive types

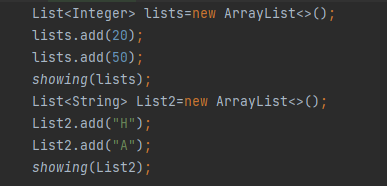


### Example 2:



We can also use wildcards when we don’t know about the parameter being passed to the methods like shown below. Here its not known wheter it’ll be an integer or string or whatever:





Don’ts for generics:

* Generics don’t work with primitive data types.
* Don’t make type parameters as static in a generic class
* Don’t instantiate a type parameter with in a class.
* Overloading is not supported in generic class