

JAVA GENERICS

Q 1 : What is the reason of including Java Generics ?

Ans : Code that uses generics has many benefits over non-generic code: Stronger type checks at compile time. A Java compiler applies strong type checking to generic code and issues errors if the code violates type safety. Fixing compile-time errors is easier than fixing runtime errors, which can be difficult to find.

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

Q 2 : Which version of Java introduce Java generics ?

Ans : java 5 .

Q 3 : What are type in Generics ?

Ans : Class and Interface .

Q 4 : How can we restricts Generics to subclass of a particular class ?

Ans : public abstract class List<T extends SomeInterface>

Q 5 : What convention follow for the type parameter in generics ?

Ans :

- 1) A single capital letter, optionally followed by a single numeral ,
- 2) A name in the form used for classes , ollowed by the capital letter T (examples: RequestT, FooBarT).

Source : <https://stackoverflow.com/questions/2900881/generic-type-parameter-naming-convention-for-java-with-multiple-chars>

Q 6 : What removed by java by introducing generics ?

Ans : compile-time type checking and removing risk of ClassCastException

Q 7 : What is wild card in Java generics ?

Ans : The ? (question mark) symbol represents wildcard element. It means represent an unknown type.

If we write <? extends Number>, it means any child class of Number.Integer, Float, double etc. Now we can call the method of Number class through any child class object.

Source : <https://www.javatpoint.com/generics-in-java>

Q 8 : Define use of unbound wildcard in java generics ?

Ans : public void printStuff(Iterable<?> stuff) {
 for (Object item : stuff) {
 System.out.println(item);
 }
}

Q 9 : What is Legacy Code ? Define the package name .

Ans : Millions of lines of code have been written in earlier versions of the language,
and they are called Legacy Code .package com.Fooblibar.widgets;

Source : <https://docs.oracle.com/javase/tutorial/extra/generics/legacy.html>

Q 10 : Define types of bounded wildcard and use of it .

Ans : 1) Upper-Bounded Wildcards

```
public static void process(List<? extends Foo> list) {
```

2)Unbounded Wildcards

```
public static void printList(List<?> list)
```

3)Lower-Bounded Wildcards

```
public static void addNumbers(List<? super Integer> list)
```

Q 11 : Can we use generics with array , How ?

Ans : Collection Framework by using list ,Map , set .

Q 12 : Define benefits of Type parameter .

Ans : Robustness , means same type can use again and again

Q 13 : What is co-varient return in java generics ;

Ans : Subtype .