

# Section 10: Java Generics

15 July 2021 19:23

## Section 10: Java Generics

### Generic Methods:-

Generics also provide compile-time type safety that allows programmers to catch invalid types at compile time.

We can write a single generic method declaration that can be called with arguments of different types. Based on the types of the arguments passed to the generic method, the compiler handles each method call appropriately. 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 ( < E > in the next example).

```
public class GenericMethodTest
{
    // generic method printArray
    public static < E > void printArray( E[] inputArray ) {
        // Display array elements
        for(E element : inputArray) {
            System.out.printf("%s ", element);
        }
        System.out.println();
    }

    public static void main(String args[]) {
        // Create arrays of Integer, Double and Character
        Integer[] intArray = { 1, 2, 3, 4, 5 };
        Double[] doubleArray = { 1.1, 2.2, 3.3, 4.4 };
        Character[] charArray = { 'H', 'E', 'L', 'L', 'O' };

        System.out.println("Array integerArray contains:");
        printArray(intArray); // pass an Integer array

        System.out.println("\nArray doubleArray contains:");
        printArray(doubleArray); // pass a Double array

        System.out.println("\nArray characterArray contains:");
        printArray(charArray); // pass a Character array
    }
}
```

Bounded Type Parameters (Example) :

```
public class MaximumTest {
    // determines the largest of three Comparable objects

    public static <T extends Comparable<T>> T maximum(T x, T y, T z) {
        T max = x; // assume x is initially the largest

        if(y.compareTo(max) > 0) {
```

```

        max = y; // y is the largest so far
    }

    if(z.compareTo(max) > 0) {
        max = z; // z is the largest now
    }
    return max; // returns the largest object
}

```

### Generic Classes :

A generic class declaration looks like a non-generic class declaration, except that the class name is followed by a type parameter section.

```

public class Box<T> {
    private T t;

    public void add(T t) {
        this.t = t;
    }

    public T get() {
        return t;
    }

    public static void main(String[] args) {
        Box<Integer> integerBox = new Box<Integer>();
        Box<String> stringBox = new Box<String>();

        integerBox.add(new Integer(10));
        stringBox.add(new String("Hello World"));

        System.out.printf("Integer Value :%d\n\n", integerBox.get());
        System.out.printf("String Value :%s\n", stringBox.get());
    }
}

```