

Java - Generics

Md. Mohsin Uddin

East West University

mmuddin@ewubd.edu

July 22, 2019

Generic Methods : Part I

```
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  
    }  
}
```

Generic Methods : Part II

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

```
System.out.println("\nArray_characterArray_contains:");  
printArray(charArray);    // pass a Character array
```

```
}  
}
```

When the above code is compiled and executed, it produces the following result:

Array integerArray contains:

1 2 3 4 5

Array doubleArray contains:

1.1 2.2 3.3 4.4

Array characterArray contains:

H E L L O

Generic Classes: Example 1 : Part I

```
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",
```

Generic Classes: Example 1 : Part II

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

When the above code is compiled and executed, it produces the following result:

| Integer Value :10 String Value :Hello World

Generic Classes: Example 2 : Part I

```
// A Simple Java program to show working of user defined  
// Generic classes
```

```
// We use < > to specify Parameter type
```

```
class Test<T>  
{  
    // An object of type T is declared  
    T obj;  
    Test(T obj) { this.obj = obj; } // constructor  
    public T getObject() { return this.obj; }  
}
```

```
// Driver class to test above
```

```
class Main  
{  
    public static void main (String[] args)  
    {  
        // instance of Integer type
```

Generic Classes: Example 2 : Part II

```
Test <Integer> iObj = new Test<Integer>(15);
System.out.println(iObj.getObject());

// instance of String type
Test <String> sObj =
    new Test<String>("GeeksForGeeks");
System.out.println(sObj.getObject());
}
}
```

When the above code is compiled and executed, it produces the following result:

```
15
GeeksForGeeks
```

Generic Classes: Example 3 : Part I

```
// A Simple Java program to show multiple  
// type parameters in Java Generics
```

```
// We use < > to specify Parameter type
```

```
class Test<T, U>
```

```
{
```

```
    T obj1; // An object of type T
```

```
    U obj2; // An object of type U
```

```
// constructor
```

```
Test(T obj1, U obj2)
```

```
{
```

```
    this.obj1 = obj1;
```

```
    this.obj2 = obj2;
```

```
}
```

```
// To print objects of T and U
```

```
public void print()
```

```
{
```


Generic Classes: Example 3 : Part II

```
        System.out.println(obj1);
        System.out.println(obj2);
    }
}

// Driver class to test above
class Main
{
    public static void main (String[] args)
    {
        Test <String , Integer> obj =
            new Test<String , Integer>(" GfG" , 15);

        obj.print();
    }
}
```

Generic Classes: Example 3 : Part III

When the above code is compiled and executed, it produces the following result:

```
| GfG  
| 15
```

References



DEITEL, Java How to Program, 11/e



Java: the complete reference, Herbert Schildt, McGraw-Hill Education Group