**Consumer Functional Interface Example In Java 8**



The Consumer is a predefined functional Interface and available in the “java.util.function” package. It consumes some item or object. It takes a single input argument and performs some operation on that argument and doesn’t return any result. If we require the implementation of a Lambda Expression with the predefined functional interface and don’t want to return any result through this implementation then we prefer [Consumer Predefined Functional Interface](https://springjava.com/core-java/consumer-functional-interface-example-java-8).

**Prototype**

public interface Consumer<T>{  
public void accept(T t);  
}

**Type Parameters**

T — It is a type of input to the operation

**Example**

import java.util.function.Consumer;public class Test { public static void main(String[] args) { Consumer < String > c = stdtName -> System.out.println(stdtName); c.accept("Peter"); c.accept("Parker"); c.accept("William"); c.accept("Smith"); c.accept("John"); c.accept("Robert"); }}

**Output**

Peter  
Parker  
William  
Smith  
John  
Robert

**Example**

import java.util.function.Consumer;import java.util.function.Function;import java.util.function.Predicate;class Student { String name; int marks; public Student(String name, int marks) { this.name = name; this.marks = marks; }}public class Test { public static void main(String[] args) { Function < Student, String > f = sdnt -> { int marks = sdnt.marks; String grades = ""; if (marks >= 90) grades = "A+"; else if (marks >= 80) grades = "A"; else if (marks >= 70) grades = "B+"; else if (marks >= 60) grades = "B"; else if (marks >= 50) grades = "C"; else if (marks >= 40) grades = "D"; else grades = "E"; return grades; }; Predicate < Student > p = sdnt -> sdnt.marks > 60; Consumer < Student > c = stu -> { System.out.println("Student Name:" + stu.name); System.out.println("Student Marks:" + stu.marks); System.out.println("Student Grade:" + f.apply(stu)); System.out.println("------------------------------"); }; Student[] s = {  
 new Student("Raghav", 100),  
 new Student("Raj", 80),  
 new Student("Varun", 70),  
 new Student("Vikas", 60),  
 new Student("Manav", 50),  
 new Student("Dinesh", 40),  
 new Student("Deepak", 30)  
 }; for (Student s1: s) { if (p.test(s1)) { c.accept(s1); } } }}

→ In the above example we used Consumer for printing Student Information and whenever we require to print student information then we use this Consumer. This is the benefit of functional programming features.

→ In the above example, we used [Predicate, Function](https://springjava.com/core-java/predicate-functional-interface-in-java-8) and Consumer.

**Output**

Student Name:Raghav  
Student Marks:100  
Student Grade:A+  
------------------------------  
Student Name:Raj  
Student Marks:80  
Student Grade:A  
------------------------------  
Student Name:Varun  
Student Marks:70  
Student Grade:B+  
------------------------------

**Consumer Chaining**

Multiple Consumers are combined together into a Consumer called **Consumer Chaining**.

**Example**

import java.util.function.Consumer;class WebSeries { String name; public WebSeries(String name) { this.name = name; }}public class Test { public static void main(String[] args) { Consumer < WebSeries > c1 = ws -> System.out.println(ws.name + " is going to release"); Consumer < WebSeries > c2 = ws -> System.out.println(ws.name + " is released and public reviews are very good"); Consumer < WebSeries > c3 = ws -> System.out.println(ws.name + " is Super Hit"); Consumer < WebSeries > cc = c1.andThen(c2).andThen(c3); WebSeries webSeries = new WebSeries("Panchayat Season2"); cc.accept(webSeries); }}

**Output**

Panchayat Season2 is going to releasePanchayat Season2 is released and public reviews are very goodPanchayat Season2 is Super Hit

**Conclusion**

This topic is explained:

• What is Consumer and how to use it?

• How to use Predicate and [Function](https://springjava.com/core-java/what-is-the-function-interface-java8)?

• How to use functional programming with the help of the Consumer?

• What is Consumer Chaining and how to use it?