# Guided LAB - 303.11.1 - Generic Method and Class

# 

# Lab Objective:

In this lab, we will demonstrate a generic method and class. By the end of this lab, learners will be able to utilize the generic method and class.

# 

# Example: Java Generics Method

We can create a method that can be used with any type of data. That method is known as the ***Generics Method***.

Create a class named **DemoClass**. As shown below, we will create a generic method in this class:

| Public class DemoClass {  // create a generics method  public <T> void genericsMethod(T data) {  System.out.println("Generics Method:");  System.out.println("Data Passed: " + data);  }  } |
| --- |

Create a class named ***myRunner***. In this class, we will invoke the generic method.

| public class myRunner {  public static void main(String[] args) { // initialize the class with Integer data  DemoClass dObj = new DemoClass();  dObj.genericsMethod(25); // passing int  dObj.genericsMethod("Per Scholas"); // passing String  dObj.genericsMethod(2563.5); // passing float  dObj.genericsMethod('H'); // passing Char  } } |
| --- |

**Run your program:**

**Output:**

Generics Method:

Data Passed: 25

Generics Method:

Data Passed: Per Scholas

Generics Method:

Data Passed: 2563.5

Generics Method:

Data Passed: H

In the above example, we have created a generic method named genericsMethod.

public <T> void genericMethod(T data) {...}

Here, the type parameter <T> is inserted after the public modifier and before the return type void.

We can call the generics method by placing the actual type <String> and <Integer> inside the bracket before the method name.

# 

# 

# 

# Example: Generic Class

A class can have more than one **type parameter**. In this case, the type parameters are separated by a comma.

For the demonstration, we will initialize two **type parameters**  in the Generic class. The names of the parameter types will be Datatypeone and DatatypeTwo, but these are only names. You are free to use “X” or “Z,” or any other identifier to name parameters.

Create a class named **GMultipleDatatype**: Write the below code.

| **public class** GMultipleDatatype <Datatypeone, DatatypeTwo> {  Datatypeone **valueOne**;  DatatypeTwo **valueTwo**;  **public** GMultipleDatatype(Datatypeone v1, DatatypeTwo v2)  {  **this**.**valueOne** = v1;  **this**.**valueTwo** = v2;  }  **public** Datatypeone getValueOne() {  **return valueOne**;  }  **public void** setValueOne(Datatypeone valueOne) {  **this**.**valueOne** = valueOne;  }  **public** DatatypeTwo getValueTwo() {  **return valueTwo**;  }  **public void** setValueTwo(DatatypeTwo valueTwo) {  **this**.**valueTwo** = valueTwo;  }  } |
| --- |

Create a class named **MyRunner** as shown below:

| **public class** MyRunner {  **public static void** main(String[] args) {  *// initialize generic class*  *// with String and Integer data*    GMultipleDatatype<String, Integer> mobj = **new** GMultipleDatatype(**"Per Scholas"**, 11025);  System.***out***.println(mobj.getValueOne());  System.***out***.println(mobj.getValueTwo());    *// initialize generic class*  *// with String and String data*  GMultipleDatatype<String, String> mobj2 = **new** GMultipleDatatype(**"Per Scholas"**, **"Non profit"**);  System.***out***.println(mobj2.getValueOne());  System.***out***.println(mobj2.getValueTwo());  }  } |
| --- |

**Run your program:**

**Output:**

Per Scholas

11025

Per Scholas

Non profit

**Submission Instructions:**

Include the following deliverables in your submission -

* + Submit your source code using the Start Assignment button in the top-right corner of the assignment page in Canvas.

**CANVAS STAFF USE ONLY: Canvas Submission Guideline:**

| **Instructions for Canvas Assignment Creation** |
| --- |
| **Assignment Name: GLAB - 303.11.1 - Generic Method and Class**  **Points:** **100**  **Assignment Group: Module 303: Java SE Review (Not Graded)**  **Display Grade As: Complete/Incomplete**  **Do not count this assignment towards the final grade: Checked**  **Submission Types: Files uploads**  **Everything else is the default.** |