4. Method References

- Sometimes, You may have method already available with some logic which you want write in the Lambda Expression. In this case, you can use existing method instead of duplicating the code again.
- You can use the Existing Method References to re-use the logic.
- Method references are a special type of lambda expressions which are used to create simple lambda expressions by referencing existing methods.
- There are 3 types of method references.
    1) Static Method Reference
    2) Instance method Reference
    3) Constructor Reference
- A new operator **::(double colon)** called as Method Reference Delimiter to specify the Method References

Static Method Reference:

- Allows to access existing Static Methods

Syntax:

```
Class::staticMethod
```

Ex:

```
Hello hello = MyInteger::findSum;
```

Instance Method Reference:

- Allows to access existing Instance or Non Static Methods

Syntax:

```
ObjRef::instanceMethod
```

Ex:

```
MyInteger myIntRef = new  MyInteger();
Hello hello2 = myIntRef::findSum;
```

Constructor Method Reference:

- Allows to access constructor of Class to Create the Object

Syntax:

```
Class::new
```

Ex:

```
Hello hello= Course::new;
```

Demo1: Files Required:

1. Hello.java	2. MyInteger.java
3. Demo1.java	

1)Hello.java

```
package com.jlcindia.demo1;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
@FunctionalInterface
public interface Hello {
	public int test(int a, int b);
}
```

2)MyInteger.java

```
package com.jlcindia.demo1;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class MyInteger {
	public static int findSum(int a, int b) {
		return a + b;
	}
}
```

3)Demo1.java

```
package com.jlcindia.demo1;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Demo1 {

public static void main(String[] args) {

Hello hello1 = (a, b) -> {
int sum = a+b;
return sum;
};

int sum1 = hello1.test(100, 50);
System.out.println("Sum : " + sum1);

Hello hello2 = MyInteger::findSum;

int sum2 = hello2.test(100, 50);
System.out.println("Sum : " + sum2);

Hello hello3 = Integer::sum;

int sum3 = hello3.test(100, 50);
System.out.println("Sum : " + sum3);

Hello hello4 = Integer::max;

int max = hello4.test(100, 50);
System.out.println("Max : " + max);

Hello hello5 = Integer::min;

int min = hello5.test(100, 50);
System.out.println("Min : " + min);

System.out.println("Done!!!");
}
}
```

Demo2: Files Required:

1. Hello.java	2. Hai.java
3. MyInteger.java	4. Demo2.java

1)Hello.java

```
package com.jlcindia.demo2;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

@FunctionalInterface
public interface Hello {
            public int test(int a, int b);
}
```

2)Hai.java

```
package com.jlcindia.demo2;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
@FunctionalInterface
public interface Hai {
    public void test(String str);
}
```

3)MyInteger.java

```
package com.jlcindia.demo2;

/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class MyInteger {
    public int findSum(int a, int b) {
        return a + b;
    }
}
```

4)Demo2.java

```
package com.jlcindia.demo2;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Demo2 {

public static void main(String[] args) {


Hello hello1 = (a, b) -> {
int sum = a+b;
return sum;
};
int sum1 = hello1.test(100, 50);
System.out.println("Sum : " + sum1);

MyInteger myIntRef = new  MyInteger();
Hello hello2 = myIntRef::findSum;

int sum2 = hello2.test(100, 50);
System.out.println("Sum : " + sum2);

Hai hai1 = (msg) -> {
System.out.println(msg);
};

hai1.test(" Hai Guys!!! ");

Hai hai2 = System.out :: println;

hai2.test(" Hey Guys !!! ");

}
}
```

Demo3: Files Required:

1. Hello.java	2. Course.java
3. Demo3.java	

1)Hello.java

```
package com.jlcindia.demo3;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
@FunctionalInterface
public interface Hello {
	public Course test(int a,String b,String c,String d);
}
```

2)Course.java

```
package com.jlcindia.demo3;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Course {
private int courseId;
private String courseName;
private String duration;
private String trainer;

public Course() {
System.out.println("Course - 0 arg Con");
}
public Course(int courseId, String courseName, String duration, String trainer) {
System.out.println("Course - 4 arg Con");
this.courseId = courseId;
this.courseName = courseName;
this.duration = duration;
this.trainer = trainer;
}
//Setters and Getters
//toString() method
}
```

3)Demo3.java

```
package com.jlcindia.demo3;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Demo3 {

public static void main(String[] args) {

//1.Lambda Style
Hello hello1 = (a, b, c, d) -> {
Course course = new Course(a, b, c, d);
return course;
};
Course course1 = hello1.test(101, "DevOps","60 Hrs","Srinivas Dande");
System.out.println(course1);

//2.Method Refernce Style
Hello hello2= Course::new;

Course course2 = hello2.test(102, "Boot - MicroServices","100 Hrs","Srinivas Dande");
System.out.println(course2);

System.out.println("Done!!!");
}
}
```

Demo4: Files Required:

1. Hello.java	2. Demo4.java

1)Hello.java

```
package com.jlcindia.demo4;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

@FunctionalInterface
public interface Hello {
        public void test(int[] arr);
}
```

2)Demo4.java

```
package com.jlcindia.demo4;

import java.util.Arrays;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */

public class Demo4 {

public static void main(String[] args) {
```

```
int myarr1[] = { 20, 40, 30, 50, 10 };

//1.Lambda Style

Hello hello1 = (arr) -> {

        for (int i = 0; i < arr.length -1; i++) {
                for (int j = i + 1; j < arr.length; j++) {
                        if (arr[i] > arr[j]) {
                        int temp = arr[i];
                        arr[i] = arr[j];
                        arr[j] = temp;
                        }
                }
        }
};


hello1.test(myarr1);

for (int x : myarr1) {
System.out.println(x);
}
System.out.println("------------------");

int myarr2[] = { 99, 88, 20, 40, 30, 50, 10 };

//2.Method Refernce Style

Hello hello2 = Arrays::sort;

hello2.test(myarr2);

for (int x : myarr2) {
System.out.println(x);
}

}
}
```

Demo5.java

```
package com.jlcindia.demo5;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Stream;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class Hello {
public static void show(int x) {
System.out.println(x);
}
}
public class Demo4 {
public static void main(String[] args) {

List<Integer> mylist = new ArrayList<>();
mylist.add(30);
mylist.add(20);
mylist.add(50);
mylist.add(10);
mylist.add(40);

Stream<Integer> mystream= mylist.stream();
mystream.forEach(Hello::show);

System.out.println("-------------------");

mylist.stream().forEach(Hello::show);  //Static Method Ref Style
System.out.println("-------------------");

mylist.stream().forEach(System.out::println);  //Instance Method Ref Style
System.out.println("-------------------");

mylist.stream().forEach( (x) -> System.out.println(x));  //Eambda Style
}
}
```

Demo6.java

```
package com.jlcindia.demo6;

import java.util.ArrayList;
import java.util.List;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
class MyNumber {
public static boolean isEven(int number) {
if (number % 2 == 0)
return true;
else
return false;
}
public static boolean isOdd(int number) {
if (number % 2 != 0)
return true;
else
return false;
}
}

public class Demo5 {
public static void main(String[] args) {
List<Integer> mylist = new ArrayList<>();
mylist.add(3);  mylist.add(2);  mylist.add(5);  mylist.add(1);  mylist.add(4);

mylist.stream()
.filter(MyNumber::isEven)
.forEach(System.out::println);

System.out.println("------------------");

mylist.stream()
.filter(MyNumber::isOdd)
.forEach(System.out::println);
System.out.println("------------------");

mylist.stream()
.filter(a -> a % 2 == 0)
.forEach(a -> System.out.println(a));
System.out.println("------------------");
mylist.stream()
.filter(a -> a % 2 != 0)
.forEach(a -> System.out.println(a));
System.out.println("-------------------");
}
}
```

Demo7.java

```
package com.jlcindia.demo7;

import java.util.*;
/*
* @Author : Srinivas Dande
* @Company: Java Learning Center
* */
public class Demo6 {
public static void main(String[] args) {

List<Integer> mylist = new ArrayList<>();
mylist.add(3);
mylist.add(2);
mylist.add(5);
mylist.add(1);
mylist.add(4);
mylist.add(6);
mylist.add(7);
mylist.add(8);

mylist.stream()
.filter(a -> a % 2 == 0)
.map(a ->  a* a)
.forEach(a -> System.out.println(a));

System.out.println("-------------------");

mylist.stream()
.filter(a -> a % 2 != 0)
.map(a ->  a* a)
.forEach(a -> System.out.println(a));
}
}
```