1) define a functional interface "First" with an abstract method "void disp1()".

define a class "Second" with a method "void fun1()"

inside main function create a method reference for "First" and invoke "fun1" of "Second"

interface First{

void disp1();

}

public class Second{

void fun1() {

System.***out***.println("Void fun1");

}

public static void main(String[] args) {

Second s = new Second();

First f = s::fun1;

f.disp1();

}

}

2) define a functional interface "Third" with an abstract method "void show(int num)".

define a class "Sample" with a method "void disp(int num)"

inside main function create a method reference for "Third" and invoke "disp" of "Sample" by passing a value "500".

interface Third{

void show(int num);

}

class Sample{

void disp(int num) {

System.***out***.println(num);

}

}

public class Second{

public static void main(String[] args) {

Sample s = new Sample();

Third t = s::disp;

t.show(500);

}

}

3) define a functional interface "Fifth" with an abstract method "int add(int a,int b)"

define a class Math1 with a method "int calc(int a,int b)"

inside main function invoke "calc" method with the help of

a) lamdba expression

b) method reference

interface Fifth{

int add(int a,int b);

}

class Math1 {

int calc(int a,int b) {

return a+b;

}

}

public class Second{

public static void main(String[] args) {

Math1 m = new Math1();

Fifth f = (a,b) -> { return a+b; };

System.***out***.println(f.add(10, 20));;

Fifth f2 = m::calc;

System.***out***.println(f.add(30, 40));;

}

}

4) define a functional interface "Sixth" with an abstract method "void disp5()".

define a class "Sample" with a static method "void fun1()"

inside main function create a method reference for "Sixth" and invoke "fun1" of "Sample"

interface Sixth{

void disp5();

}

public class Sample{

void fun1() {

System.***out***.println("fun1");

}

public static void main(String[] args) {

Sample s = new Sample();

Sixth s2 = s::fun1;

s2.disp5();

}

}

5) define a functional interface "Emp" with an abstract method "void show()".

define a class "MyClass" with default constructor with the message "inside default constructor of MyClass".

inside main function instantiate "MyClass" using method reference to "Emp"

interface Emp{

void show();

}

class MyClass{

MyClass(){

System.***out***.println("inside default constructor of MyClass");

}

}

public class Sample{

public static void main(String[] args) {

Emp e = MyClass::new;

e.show();

}

}

6) define a functional interface "Seventh" with an abstract method "void fun(int k)".

define a class "Sample" with parameterized constructor with the message "inside parameterized constructor of Sample\t"+k .

inside main function instantiate "Sample" by passing value "200" using method reference to "Seventh"

interface Seventh{

void fun(int k);

}

public class Sample{

Sample(int k){

System.***out***.println("inside parameterized constructor of Sample \t"+k);

}

public static void main(String[] args) {

Seventh s = Sample::new;

s.fun(200);

}

}

7) define a class "Student" with name and age as member variables, parameterized constructor , setters , getters and "toString" method.

define a functional interface "MyInterface" with an abstract method "void show(Student ref)"

inside main function

a) create an instance of "Student" class by passing "Vijay" and '20' values.

b) create method reference for "MyInterface" and display the instance of "Student" class created above.

class Student{

private int age;

private String name;

Student(String name){

this.name = name;

}

void set(int age) {

this.age = age;

}

void get() {

System.***out***.println("Name :"+name);

System.***out***.println("Age : "+age);

}

public String toString() {

return "name : "+name+" age : "+age;

}

}

interface MyInterface{

void show(Student ref);

}

public class Sample{

public static void main(String[] args) {

Student s = new Student("Vijay");

s.set(20);

s.get();

MyInterface m = System.***out***::println;

m.show(s);

}

}