**Java-8 – Invoking Lambda Expression Using Functional Interface:**

* **Without Lambda Expression:**

interface Interf{

public void m1(int a, int b);

}

class Demo implements Interf{

public void m1(int a, int b){

System.out.println(“The sum:”+ (a+b));

}

}

class Test1{

public static void main(String[] args){

Interf i = new Demo();

i.add(10, 20);

i.add(100, 200);

}

}

* **With Lambda Expression:**

interface Interf{

public void m1(int a, int b);

}

class Demo{

public static void main(String[] args){

Interf i = (a, b) -> System.out.println(“The sum:”+(a+b));

i.m1(10, 20);

i.m1(100, 200);

}

}

* **Without Lambda Expression:**

interface Interf{

public int getLength(String s);

}

class Demo implements Interf{

public int getLength(String s){

return s.length();

}

}

class Test{

public static void main(String[] args){

Interf i = new Demo();

System.out.println(i.getLength(“Hello”));

System.out.println(i.getLength(“Without lambda expression”));

}

}

* **With Lambda Expression:**

interface Interf{

public int getLength(String s);

}

class Test{

public static void main(String[] args){

Interf i = s -> s.length();

System.out.println(i.getLength(“Hello”));

System.out.println(i.getLength(“With lambda expression”));

}

}

* **Without Lambda Expression:**

interface Interf{

public int squareIt(int x);

}

class Demo implements Interf{

public int squareIt(int x){

return x \* x;

}

}

class Test{

public static void main(String[] args){

Interf i = new Demo();

System.out.println(i.squareIt(2));

System.out.println(i.squareIt(8));

}

}

* **With Lambda Expression:**

interface Interf{

public int squareIt(int x);

}

class Test{

public static void main(String[] args){

Interf i = x -> return x \* x;

System.out.println(i.squareIt(2));

System.out.println(i.squareIt(8));

}

}