

## Function (Greet Test)

```
Greeting.java x
Source History
1 package func;
2
3 @FunctionalInterface
4 public interface Greeting {
5     public void greet();
6     default void x() {}
7     default void y() {}
8 }
9
```

```
Hello.java x
Source History
1 package func;
2
3
4 public class Hello implements Greeting{
5
6     @Override
7     public void greet() {
8         System.out.println("Hello");
9     }
10
11 }
```

```
Goodbye.java x
Source History
1 package func;
2
3 @FunctionalInterface
4 public interface Goodbye {
5     public void bye (String name);
6 }
```

```
Welcome.java x
Source History
1 package func;
2
3 @FunctionalInterface
4 public interface Welcome<T> {
5     public String hi();
6 }
```

```
ShakeHand.java x
Source History
1 package func;
2
3 public interface ShakeHand {
4     public String shake(String fname ,String name);
5 }
```

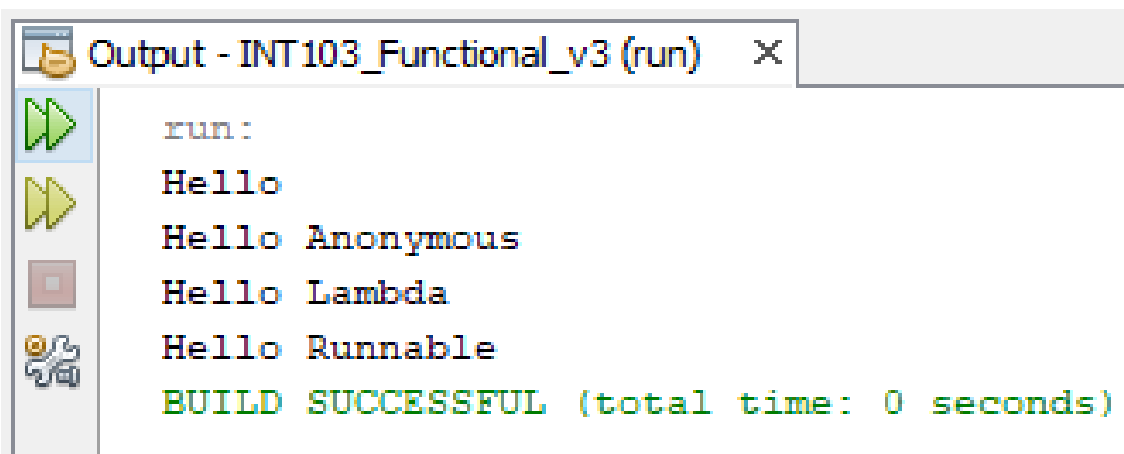
**greetTest();**

```
private static void greetTest() {
    Greeting g1 = new Hello();
    g1.greet();

    Greeting g2 = new Greeting() {
        public void greet() {
            System.out.println("Hello Anonymous");
        }
        public void x() {}
    };
    g2.greet();

    Greeting g3 = () -> System.out.println("Hello Lambda");
    g3.greet();

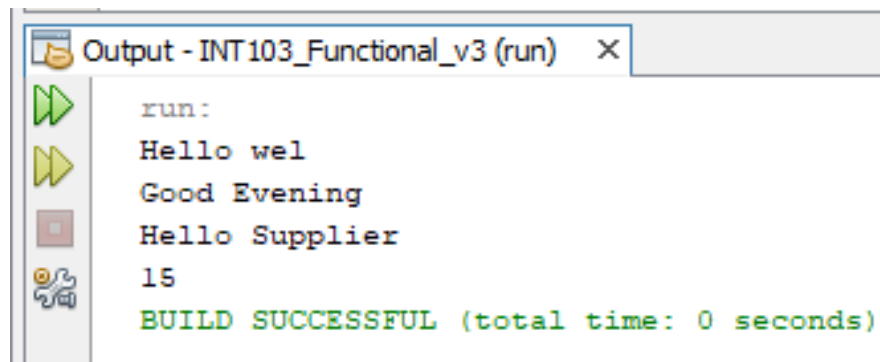
    Runnable r = () -> System.out.println("Hello Runnable");
    r.run();
}
```



```
Output - INT103_Functional_v3 (run) X
run:
Hello
Hello Anonymous
Hello Lambda
Hello Runnable
BUILD SUCCESSFUL (total time: 0 seconds)
```

## welcomeTest(); + Supplier

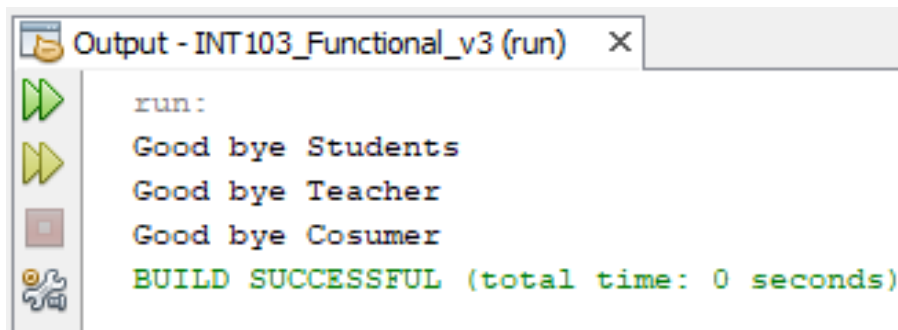
```
private static void welcomeTest(){
    //Welcome w1<String> = () -> "Hello wel";
    Welcome w1 = () -> "Hello wel";
    Welcome w2 = () -> { return "Good Evening"; };
    //
    //    Welcome w2 = () -> {
    //        String x = "Functional";
    //        return "Good Evening";
    //    };
    System.out.println(w1.hi() + '\n' + w2.hi());
    Supplier<String> s = () -> "Hello Supplier";    //เป็น Supplier<T> เพราะอะไร
    Supplier<Integer> i = () -> 15;                //input(paramiter) แต่ return result ได้
    System.out.println(s.get());                    //excuteด้วย get()
    System.out.println(i.get());
}
```



```
Output - INT103_Functional_v3 (run) X
run:
Hello wel
Good Evening
Hello Supplier
15
BUILD SUCCESSFUL (total time: 0 seconds)
```

## goodByeTest(); + Consumer

```
private static void goodByeTest() {  
    Goodbye g1 = x -> System.out.println("Good bye " + x);  
    g1.bye("Students");  
    Goodbye g2 = x -> System.out.println("Good bye " + x);  
    g2.bye("Teacher");  
  
    Consumer<String> c = x -> System.out.println("Good bye " + x); //Consumer รับค่า input แต่ไม่ return  
    c.accept("Cosumer"); //excute คำย accept()  
}
```



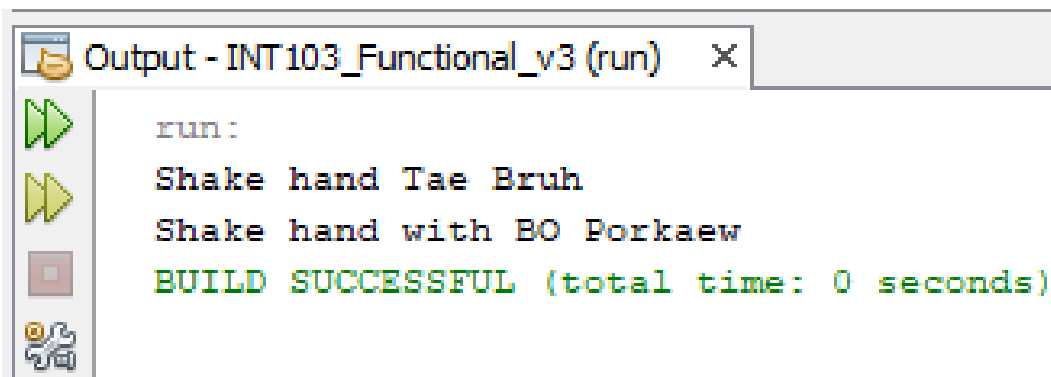
The screenshot shows an IDE output window titled "Output - INT103\_Functional\_v3 (run)". The output text is as follows:

```
run:  
Good bye Students  
Good bye Teacher  
Good bye Cosumer  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The output window includes a toolbar with icons for running, stepping through, and other debugging actions.

## shakeHandTest(); + BinaryOperator

```
private static void shakeHandTest(){  
    ShakeHand s = (f,l) -> f + ' ' + l;  
    System.out.println("Shake hand " + s.shake("Tae" ,"Bruh"));  
    BinaryOperator<String> b = (f,l) -> f + ' ' + l;  
    System.out.println("Shake hand with " + b.apply("BO", "Porkaew"));  
}
```



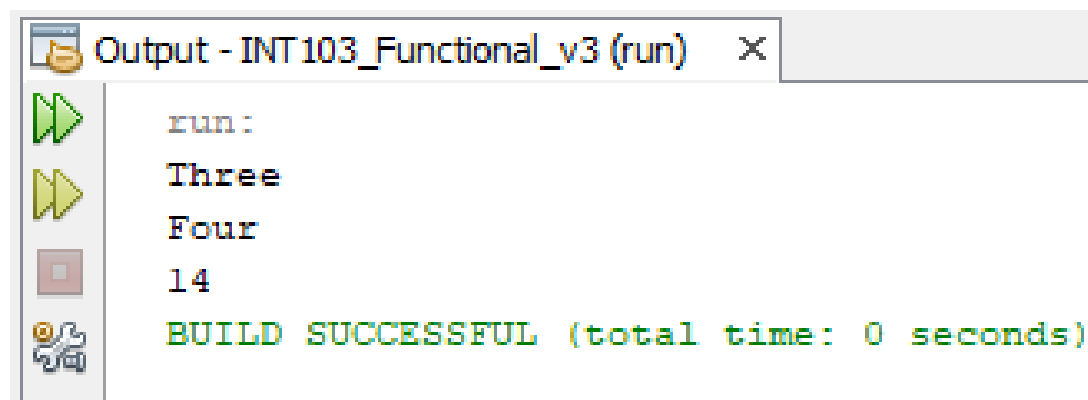
The screenshot shows the 'Output' window of an IDE, titled 'Output - INT103\_Functional\_v3 (run)'. The window contains the following text:

```
run:  
Shake hand Tae Bruh  
Shake hand with BO Porkaew  
BUILD SUCCESSFUL (total time: 0 seconds)
```

The output window includes a vertical toolbar on the left with icons for running, stepping through code, and other debugging actions.

## streamTest(); + Stream

```
private static void streamTest() {  
    String[] s = {"One", "Two", "Three", "Four", "Five"};  
  
    //      Stream<String> x = Stream.of(s);           เขียนแบบเต็ม  
    //      x  
    //          .filter(a->a.contains("r"))  
    //          .forEach(System.out::println);  
  
    Stream.of(s)                                     //เขียนแบบย่อ  
        .filter(a->a.contains("r"))  
        .forEach(System.out::println);  
    //      .forEach(x -> System.out.println(x));  
    int out = Stream.of(s)  
        .map(String::length)  
        .filter(a->a<5)  
        .reduce(0, (q, e) -> q + e);  
    System.out.println(out);  
}
```



```
Output - INT103_Functional_v3 (run) X  
run:  
Three  
Four  
14  
BUILD SUCCESSFUL (total time: 0 seconds)
```