Function (Greet Test)

Greeting.java

```java
package func;

@FunctionalInterface
public interface Greeting {
    public void greet();
    default void x() {}
    default void y() {}
}
```

Hello.java

```java
package func;


public class Hello implements Greeting{

    @Override
    public void greet() {
        System.out.println("Hello");
    }

}
```

Goodbye.java

```java
package func;

@FunctionalInterface
public interface Goodbye {
    public void bye (String name);
}
```

Welcome.java

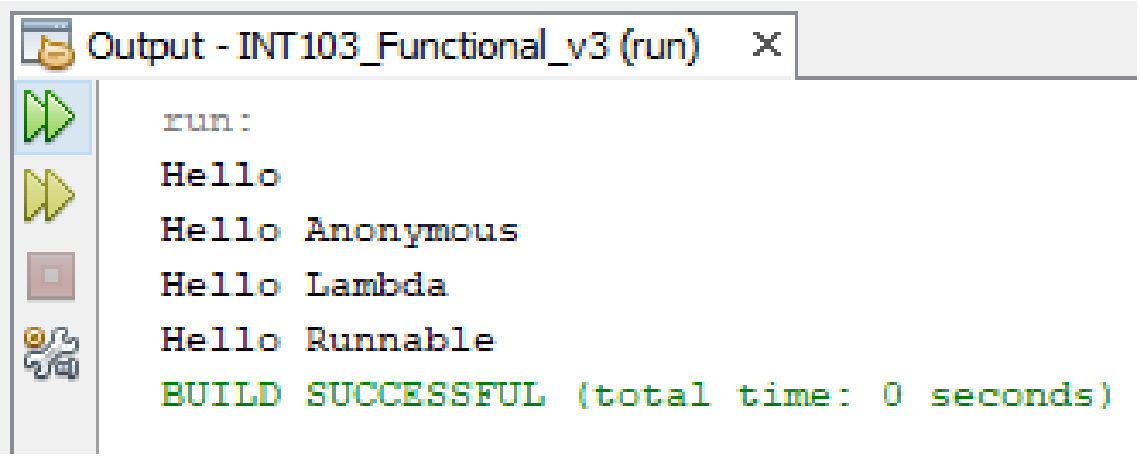
```java
package func;

@FunctionalInterface
public interface Welcome<T> {
    public String hi();
}
```

ShakeHand.java

```java
package func;

public interface ShakeHand {
    public String shake(String fname ,String name);
}
```

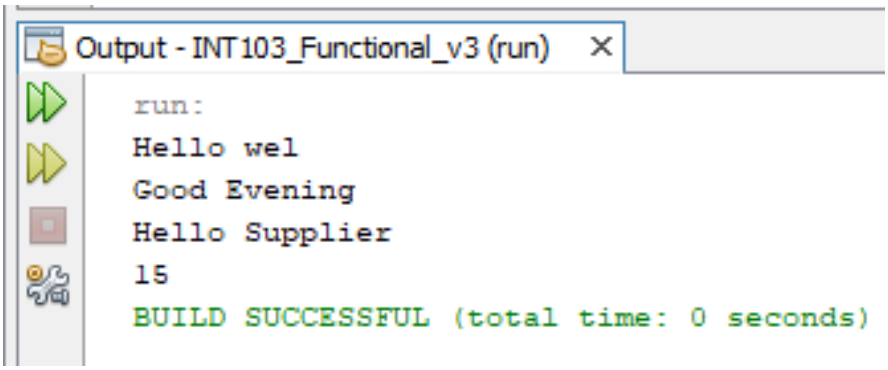
greetTest();

```
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();
}
```



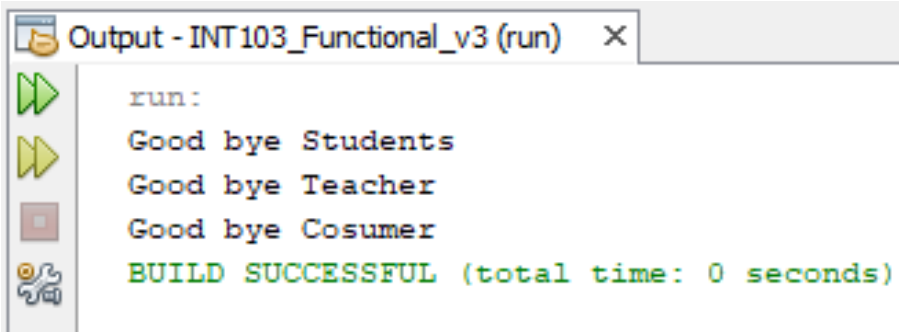
welcomeTest(); + Supplier

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



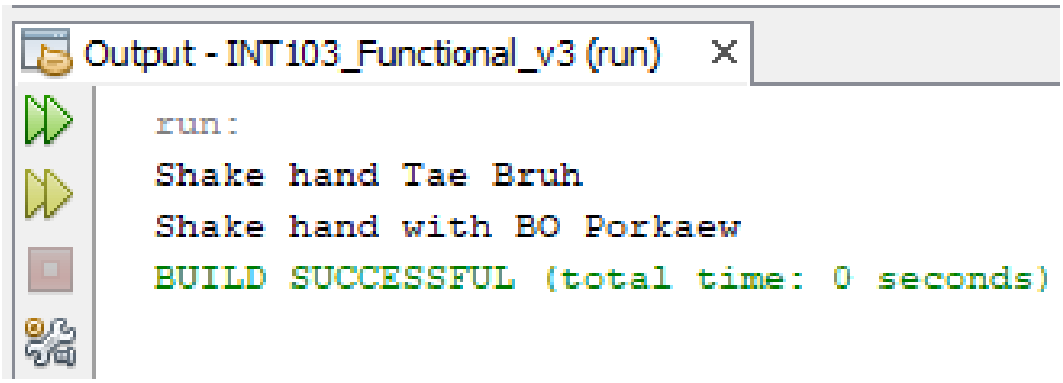
goodByeTest(); + Consumer

```
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");                                            //excuteด้วย accept()
}
```



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"));
}
```



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);
    }
}
```