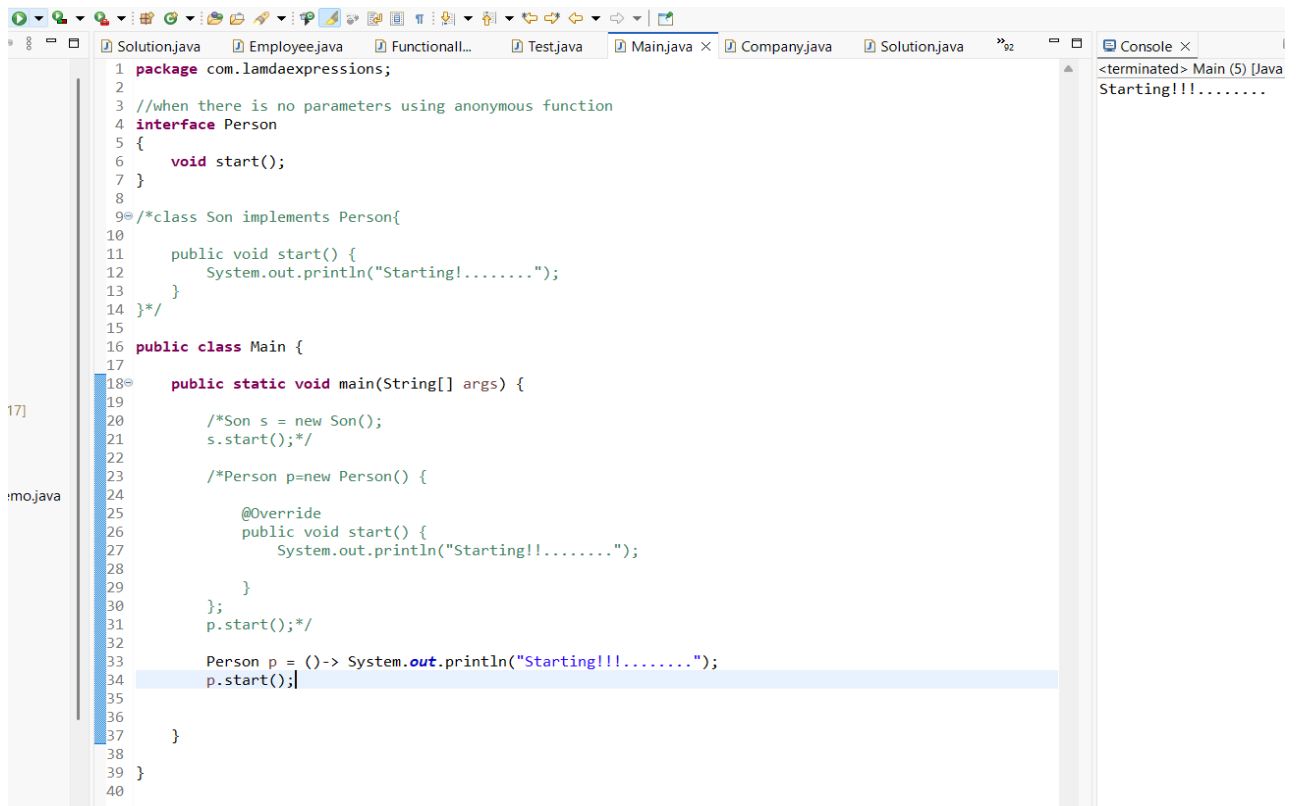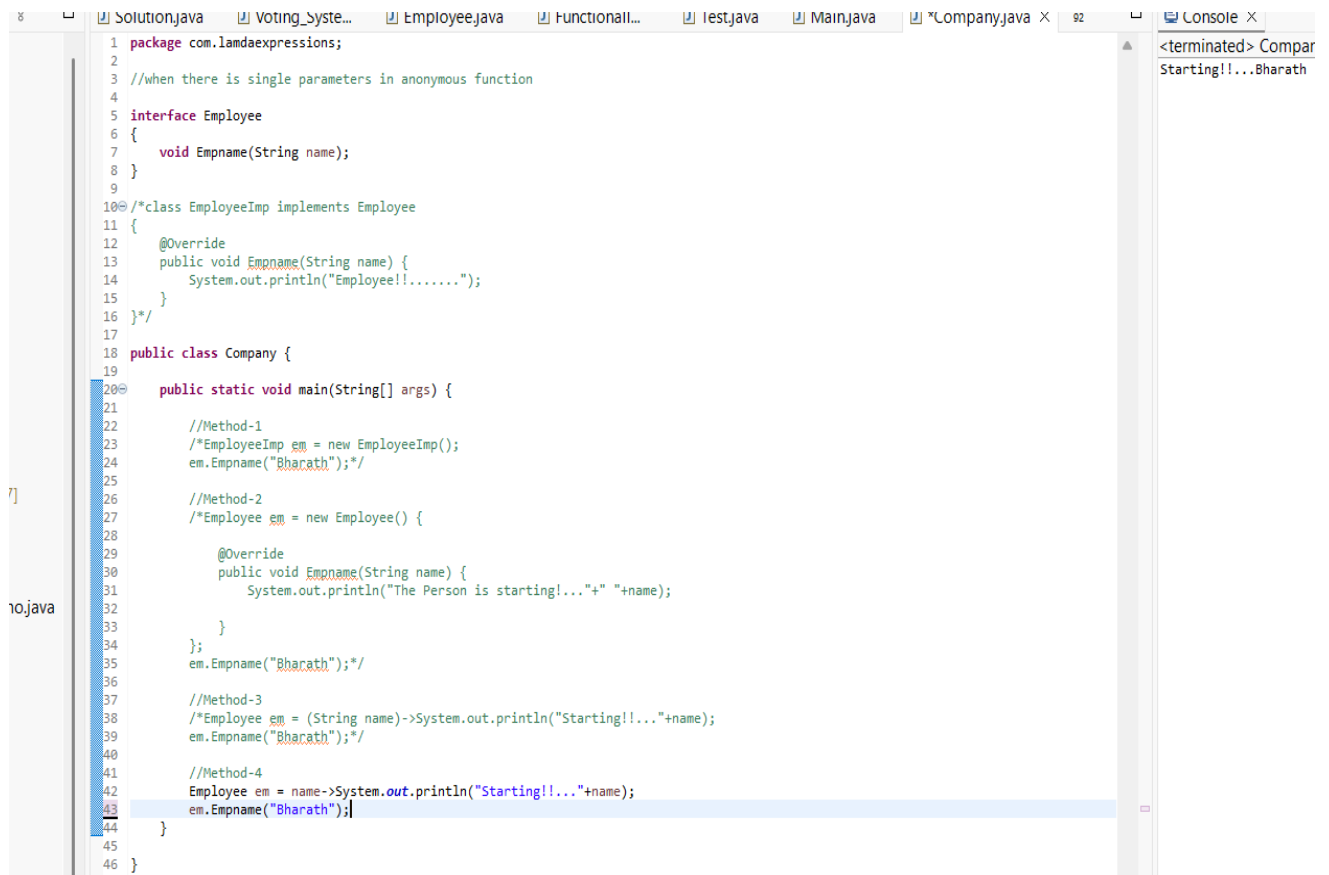# ASSIGNMENT-20

BHARATH KUMAR G R| Batch: A4| Date:30/06/2025|VSN:VS4U1298

```java
package com.lamdaexpressions;

//when there is no parameters using anonymous function
interface Person
{
    void start();
}
/*class Son implements Person{

    public void start() {
        System.out.println("Starting!........");
    }
}*/

public class Main {

    public static void main(String[] args) {

        /*Son s = new Son();
        s.start();*/

        /*Person p=new Person() {

            @Override
            public void start() {
                System.out.println("Starting!!........");
            }
        };
        p.start();*/

        Person p = ()-> System.out.println("Starting!!!........");
        p.start();

    }
}
```

Console: `<terminated> Main (5) [Java` `Starting!!!........`

```java
package com.lamdaexpressions;

//when there is single parameters in anonymous function

interface Employee
{
    void Empname(String name);
}
/*class EmployeeImp implements Employee
{
    @Override
    public void Empname(String name) {
        System.out.println("Employee!!.......");
    }
}*/

public class Company {

    public static void main(String[] args) {

        //Method-1
        /*EmployeeImp em = new EmployeeImp();
        em.Empname("Bharath");*/

        //Method-2
        /*Employee em = new Employee() {

            @Override
            public void Empname(String name) {
                System.out.println("The Person is starting!..."+" "+name);
            }
        };
        em.Empname("Bharath");*/

        //Method-3
        /*Employee em = (String name)->System.out.println("Starting!!..."+name);
        em.Empname("Bharath");*/

        //Method-4
        Employee em = name->System.out.println("Starting!!..."+name);
        em.Empname("Bharath");
    }
}
```

Console: `<terminated> Compar` `Starting!!...Bharath`

```java
package com.lamdaexpressions;

//when there is two parameters in using anonymous Method/function

interface Car
{

    void display(int a,int b);
}

public class Solution {

    public static void main(String[] args) {

        //method-1 to print
        /*Car c = new Car() {

            @Override
            public void display(int a,int b) {
                System.out.println("The number is"+a+ " " +b);
            }
        };
        c.display(50, 60);*/

        //method-2 to print
        /*Car c = (a,b)->System.out.println("The number is "+"a is :"+a+ " "+"b is:"+
        c.display(100,200);*/

        //method-3 to print
        Car c = (int a, int b)->System.out.println(a+b);
        c.display(100, 200);


    }
}
```

```java
package com.lamdaexpressions;

//lambda expression for return type method

interface A
{

    int add(int a,int b);
}

public class Test {

    public static void main(String[] args) {

        /*A  a = new A() {

            @Override
            public int add(int a, int b) {

                return a+b;
            }
        };
        int result=a.add(10, 20);
        System.out.println(result);*/

        A  aa = (a,b)-> a+b;
        int res = aa.add(10, 20);
        System.out.println(res);



    }
}
```