

## Lamda\_SquareExample

### Lamda Expression example for square of int :

```
package com.technicalstack.basic;

@FunctionalInterface
interface Interfun{

    public int square(int n);

}

public class Lamda_SquareExample {

    public static void main(String[] args) {
        Interfun i = n->(n*n);
        System.out.println("Square of 2 is :"+i.square(2));

    }

}
```

Output:

Square of 2 is :4

now suppose we put parenthesis in the body as below and try to compile the code using command prompt,

```
Interfun i = n->{(n*n)};
you will get compilation error as below:
basic>javac Lamda_SquareExample.java
Lamda_SquareExample.java:13: error: not a statement
Interfun i = n->{(n*n)};
```

```
Lamda_SquareExample.java:13: error: ';' expected
Interfun i = n->{(n*n)};
```

2 errors

so if we are putting parenthesis then return statement and ; needs to be added.

```
package com.technicalstack.basic;

@FunctionalInterface
interface Interfun{

    public int square(int n);

}

public class Lamda_SquareExample {

    public static void main(String[] args) {
        // Interfun i = n-> n*n;
```

```
Interfun i = n->{return n*n};  
System.out.println("Square of 2 is :"+i.square(2));  
  
}
```

```
}
```

### **Important:**

Below are the various scenarios:

```
Interf i = n->n*n ;//valid
```

```
Interf i = (n)->n*n ;//valid
```

```
Interf i = n->return n*n ;//invalid as return is valid only inside curly braces
```

```
Interf i = n->{retrun n*n ;} ;//valid
```

```
Interf i = n->{retrun n*n ;} //invalid,syntax error
```

```
Interf i = n->{n*n;} ;//invalid as inside curly braces return is mandatory to provide.
```