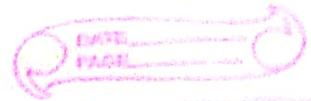


Lambda expression available  
since JDK 8.



\* Lambda Expression :

- no byte code created, reduces size of code.

- can only apply on functional interface, that is interface with single method.
- ~~same as~~ inline function of C++.

```
interface LE1 {  
    void show();  
}
```

```
class LE1Demo {
```

```
    public static void main (String[] args) {
```

```
        LE1 obj = () -> {
```

```
            System.out.println("Lambda Expression first example");
```

```
        };
```

```
        obj.show();
```

```
    }
```

```
}
```

———— X ————

in parametrized methods we don't need to initialize variable as compiler will map and identify the type of variable automatically.

if we initialize the variable an error is generated.

```
interface LE2 {
```

```
    void printSeries (int n);
```

```
}
```

```
class LE2Demo {
```

```
    public static void main (String[] args) {
```

```
        LE2 obj = (a) -> {
```

for (int i=1; i<=a; i++) {  
 System.out.println(i);  
}

obj.printSeries(10);  
}

}

---

interface LE3 {  
 int fact(int n);  
}

class LE3Demo {

public static void main(String[] args) {

LE3 obj = (a) -> {

int f=1;

for (int i=1; i<=a; i++) {

f = f \* i;

}

return f;

};

System.out.println(obj.fact(5));

}

}

---

P.T.O.