Lambda Expressions

Lambda Expressions are anonymous function, which doesn't have

- Any name
- Any return type
- Any Modifier

Steps to make any function a Lambda expression

- Remove modifier
- Remove Return type
- Remove Method name
- Place arrow

Code :

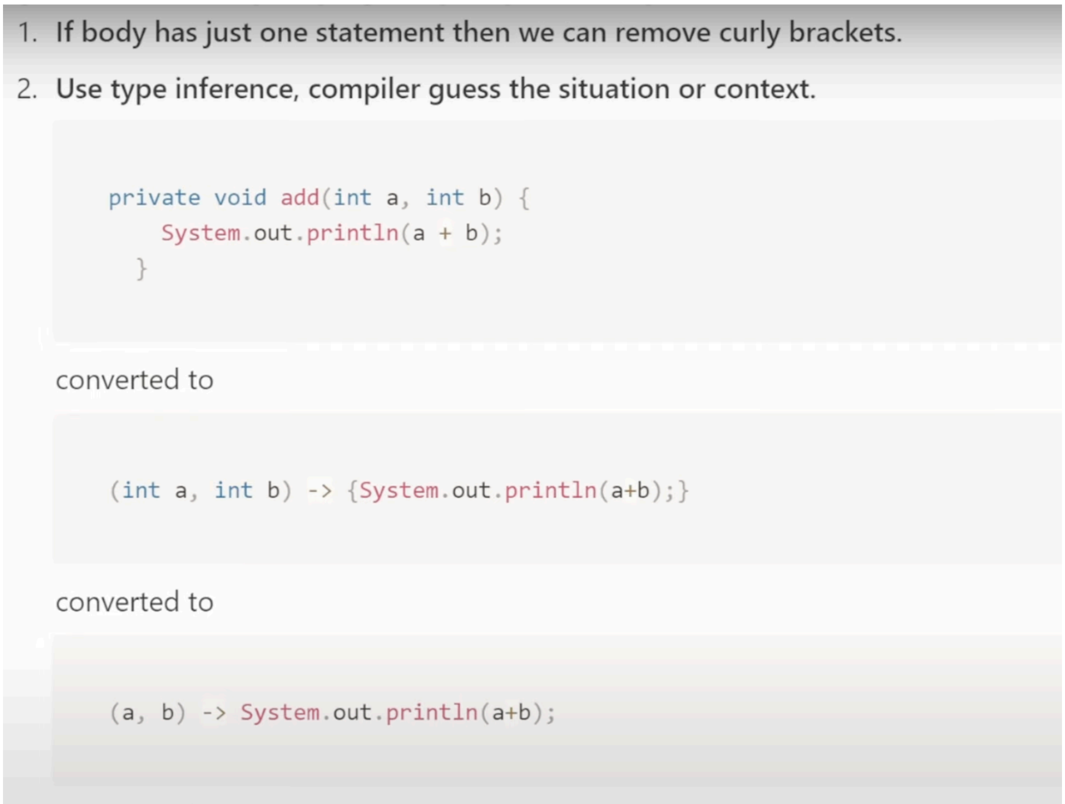
```
Private void Hello(){
        System.out.Println("Hello World");
}

Private void Add(int a , int b){
        System.out.Println(a+b);
}
```

Lambda Expression Code :

- () -> { System.out.Println("Hello World"); }
- (int a , int b) -> { System.out.Println(a+b); }
- (String str) -> { System.out.Println(str.length()); }

Properties of Lambda Expression



1. If body has just one statement then we can remove curly brackets.
2. Use type inference, compiler guess the situation or context.

```
private void add(int a, int b) {
    System.out.println(a + b);
  }
```

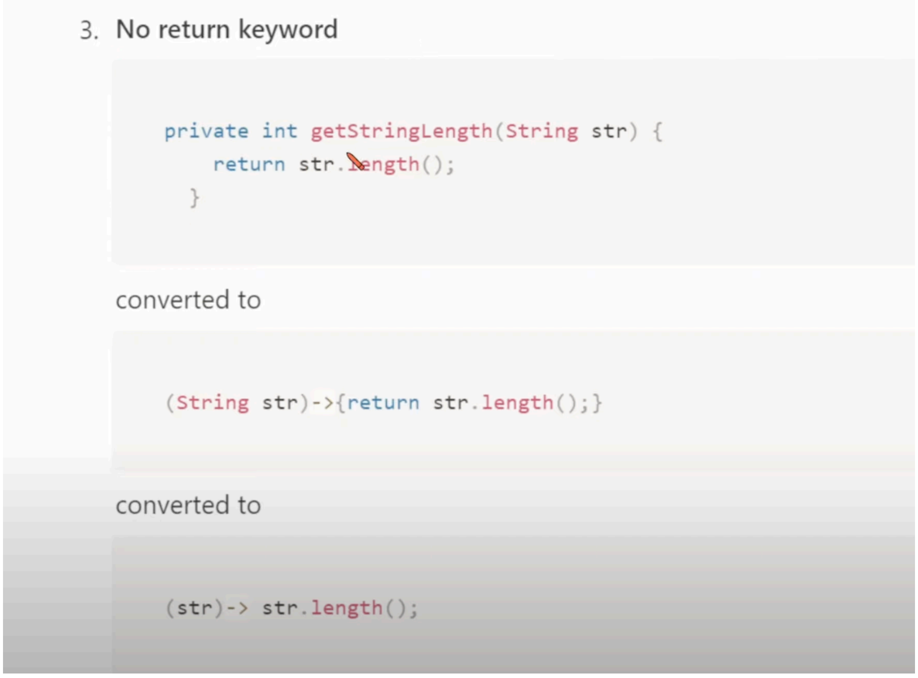
converted to

```
(int a, int b) -> {System.out.println(a+b);}
```

converted to

```
(a, b) -> System.out.println(a+b);
```

Caption



3. No return keyword

```
private int getStringLength(String str) {
    return str.length();
  }
```

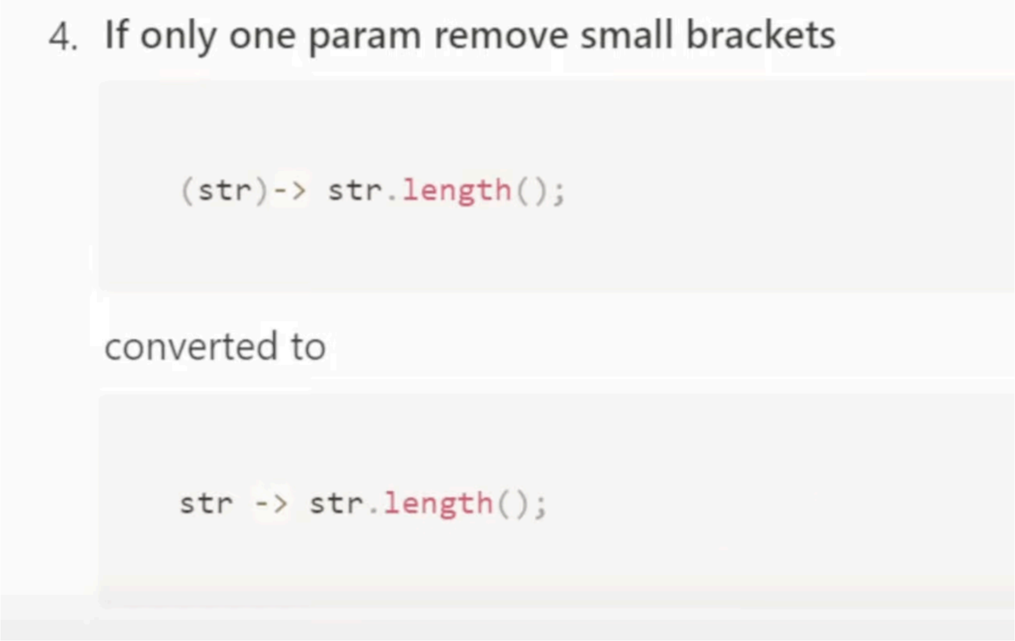
converted to

```
(String str)->{return str.length();}
```

converted to

```
(str)-> str.length();
```

Caption



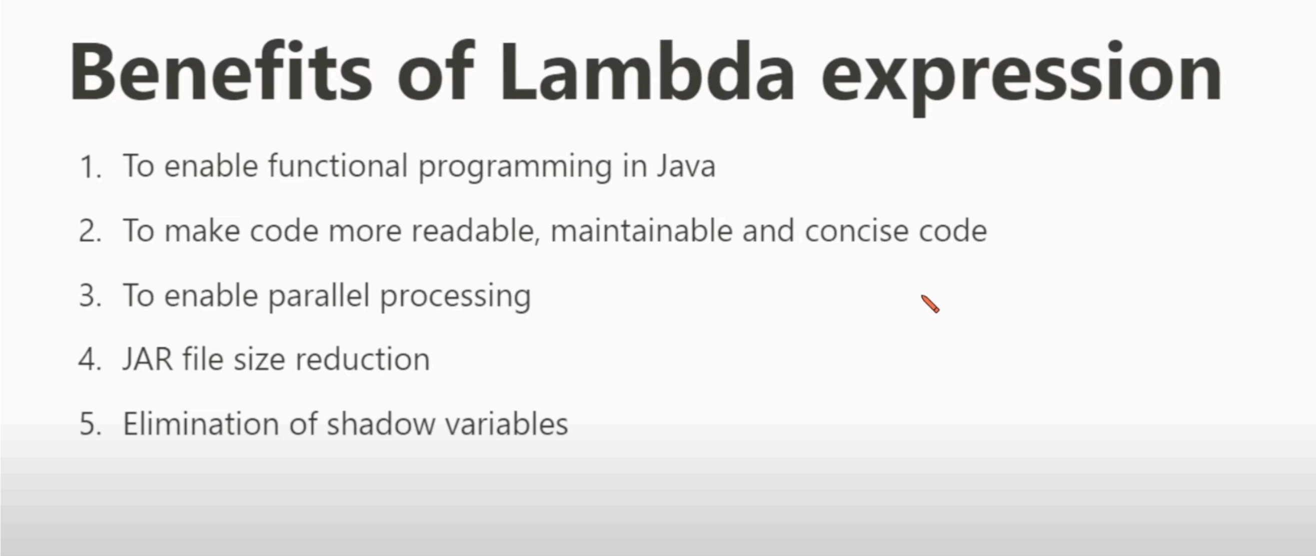
4. If only one param remove small brackets

```
(str)-> str.length();
```

converted to

```
str -> str.length();
```

Caption



Benefits of Lambda expression

1. To enable functional programming in Java
2. To make code more readable, maintainable and concise code
3. To enable parallel processing
4. JAR file size reduction
5. Elimination of shadow variables

Benefits of Lambda expression