



Functions

- Functions are exactly same as predicates except that functions can return any type of result but function should (can) return only one value and that value can be any type as per our requirement.
- To implement functions oracle people introduced Function interface in 1.8 version.
- Function interface present in *Java.util.function* package.
- Functional interface contains only one method i.e., `apply()`

```
interface function(T,R) {  
    public R apply(T t);  
}
```

Assignment: Write a function to find length of given input string.

Ex:

```
1) import Java.util.function.*;  
2) class Test {  
3)     public static void main(String[] args) {  
4)         Function<String, Integer> f = s -> s.length();  
5)         System.out.println(f.apply("Durga"));  
6)         System.out.println(f.apply("Soft"));  
7)     }  
8) }
```

Note: Function is a functional interface and hence it can refer lambda expression.



Differences between predicate and function

| Predicate | Function |
|--|--|
| To implement conditional checks We should go for predicate | To perform certain operation And to return some result we Should go for function. |
| Predicate can take one type Parameter which represents Input argument type. Predicate <T> | Function can take 2 type Parameters . First one represent Input argument type and Second one represent return Type. Function <T,R> |
| Predicate interface defines only one method called test() | Function interface defines only one Method called apply() . |
| public boolean test (T t) | public R apply (T t) |
| Predicate can return only boolean value. | Function can return any type of value |

Note: Predicate is a boolean valued function and(), or(), negate() are default methods present inside Predicate interface.