AGENDA:

Lambda and Streams

1> Functional Interfaces

2> Lambda

3> Streams → map
→ filter

INTERFACE

container for abstract methods.

```
public interface fun {

        void a();
        void b();

}
```

Q1 : Can interfaces only hold abstract methods?

→ Before Java 8
↓
YES

→ default methods are now allowed

Functional Interface:

Interface with SINGLE ABSTRACT METHOD

{ It can have default methods }

```
interface {

        void fun1( );
        default fun2( ) . .
        default fun3( )
}
```

Only single abstract method

Can I add more methods to this interface?

Yes.

@FunctionalInterface

- not mandatory but good to have
- compile time check to make sure interface only has a single abstract method

Have you ever worked with functional interfaces ?

1> Runnable ← run()
2> Callable ← → call()
3> Comparator ←
4> Comparable
5> Iterable → iterator()

Consumer<T>
↓
void accept(T t)
↓
This method can do some operation on a parameter and return nothing

BiConsumer<T, U>
↓
void accept(T, U)

Predicate<T>
↓
boolean test(T t)

Similarly → biPredicate

Callable

Function < T, U >

U apply (T t)

↓ ↑

BiFunction <T, U, V>

V apply (T t, U u

10:[illegible]