Optional

Values

APPLICATION PROGRAMMING IN JAVA

class Optional<T>

- Safer alternative to null
  - Producer:

```
Optional<String>    retVal;
if ( str == null )
    retVal = Optional.empty();
else
    retVal = Optional.of( str );
```

  - Consumer:

```
if ( opt.isPresent() )
    System.out.println( opt.get() );
else
    System.out.println( "(empty)" );
```

Common Methods

- static Optional<T> empty()
  Returns an empty Optional<T>

- static Optional<T> of(T value)
  Returns a non-empty optional

- T get()
  Returns value if present, else throws exception

- void ifPresent(Consumer<? super T> consumer)
  Executes *consumer* if non-empty

- boolean isPresent()
  Returns true if non-empty

- *See Java API documentation*

Optional Examples

- Given:

```
private String  str = null;
private Integer num = null;
private Boolean boo = null;

public Optional<String> getStr()
public Optional<Integer> getNum()
public Optional<Boolean> getBoo()
```

Optional Example 1

```
public Optional<String> getStr()
{
    Optional<String>    retVal;
    if ( str == null )
        retVal = Optional.empty();
    else
        retVal = Optional.of( str );

    return retVal;
```

Optional Example 1

```
    Optional<String> strOpt = getStr();
    if ( opt.isPresent() )
        System.out.println( opt.get() );
    else
        System.out.println( "(empty)" );
```

```
public void printVal( Optional<?> opt )
{
    if ( opt.isPresent() )
        System.out.println( opt.get() );
    else
        System.out.println( "(empty)" );
}
```

Optional Example 2

```
Optional<String>    optStr  = getStr();
Optional<Integer>   optNum  = getNum();
Optional<Boolean>   optBoo  = getBoo();

optStr.ifPresent( s -> System.out.println( s ) );
optNum.ifPresent( i -> System.out.println( i ) );
optBoo.ifPresent( b -> System.out.println( b ) );
```

Optional Example 3

```
String  myStr   = getStr().orElse( "empty" );
System.out.println( myStr );
System.out.println( getNum().orElse( -1 ) );
System.out.println( getBoo().orElse( false ) );
```