**Java 8 - Optional Class**

Optional is a container object used to contain not-null objects. Optional object is used to represent null with absent value. This class has various utility methods to facilitate code to handle values as ‘available’ or ‘not available’ instead of checking null values.

Following is the Class declaration for java.util.Optional<T> class −

**public final class Optional<T> extends Object**

**Class Methods and Method & Description**

**1. static <T> Optional<T> empty()**

Returns an empty Optional instance.

**2. boolean equals(Object obj)**

Indicates whether some other object is "equal to" this Optional.

**3. Optional<T> filter(Predicate<? super <T> predicate)**

If a value is present and the value matches a given predicate, it returns an Optional describing the value, otherwise returns an empty Optional.

**4. <U> Optional<U> flatMap(Function<? super T,Optional<U>> mapper)**

If a value is present, it applies the provided Optional-bearing mapping function to it, returns that result, otherwise returns an empty Optional.

**5. T get()**

If a value is present in this Optional, returns the value, otherwise throws NoSuchElementException.

**6. int hashCode()**

Returns the hash code value of the present value, if any, or 0 (zero) if no value is present.

**7. void ifPresent(Consumer<? super T> consumer)**

If a value is present, it invokes the specified consumer with the value, otherwise does nothing.

**8. boolean isPresent()**

Returns true if there is a value present, otherwise false.

**9. <U>Optional<U> map(Function<? super T,? extends U> mapper)**

If a value is present, applies the provided mapping function to it, and if the result is non-null, returns an Optional describing the result.

**10. static <T> Optional<T> of(T value)**

Returns an Optional with the specified present non-null value.

**11. static <T> Optional<T> ofNullable(T value)**

Returns an Optional describing the specified value, if non-null, otherwise returns an empty Optional.

**12. T orElse(T other)**

Returns the value if present, otherwise returns other.

**13. T orElseGet(Supplier<? extends T> other)**

Returns the value if present, otherwise invokes other and returns the result of that invocation.

**14. <X extends Throwable> T orElseThrow(Supplier<? extends X> exceptionSupplier)**

Returns the contained value, if present, otherwise throws an exception to be created by the provided supplier.

**15. String toString()**

Returns a non-empty string representation of this Optional suitable for debugging.

**16. This class inherits methods from the fol lowing class −**

java.lang.Object