**JAVA 8-FEATURES**

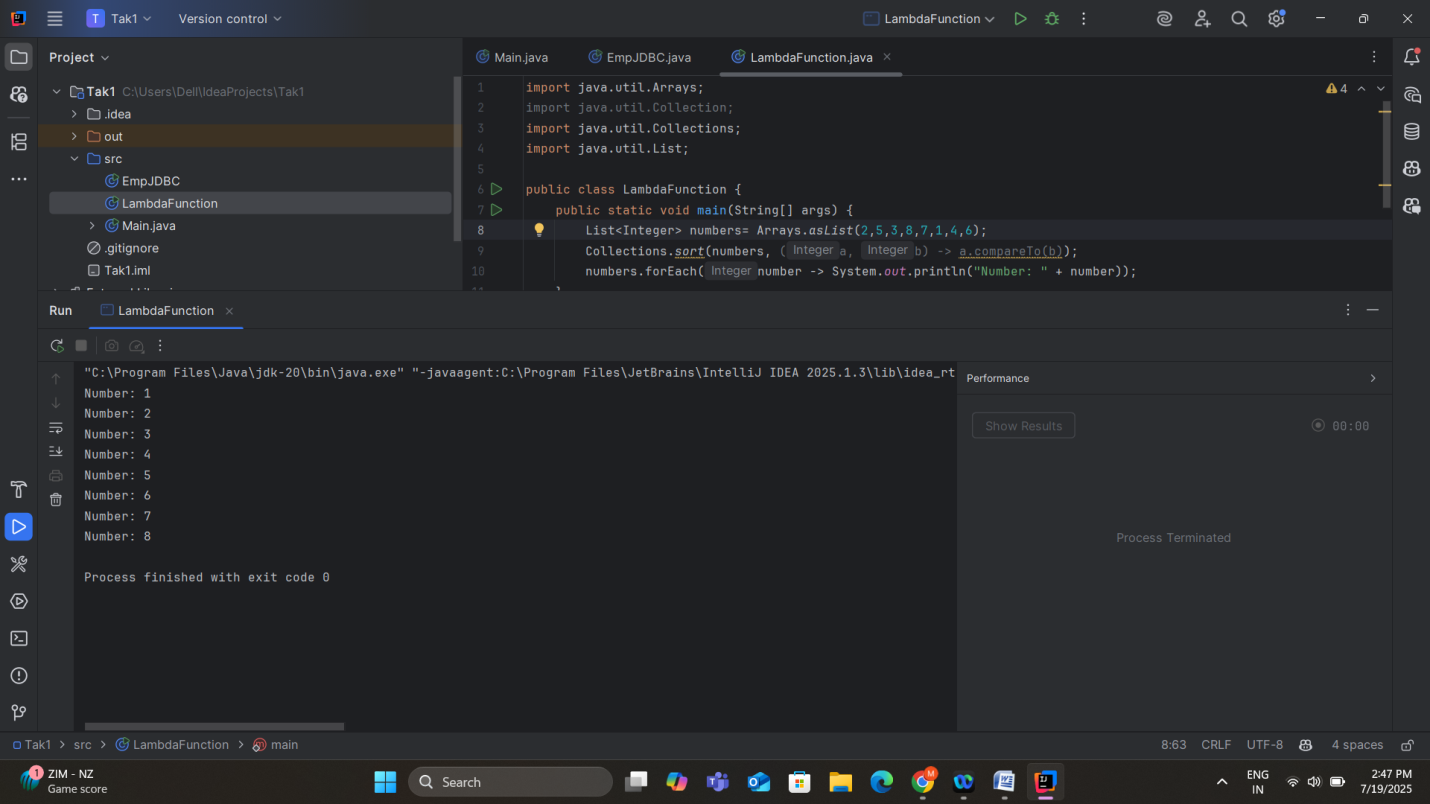
1. **List the features of Java 8**

* Lambda Expressions-----(a,b)->a+b
* Functional Interfaces----Runnable,Annotations
* Stram API
* Default mathods in interface
* Method references
* New date and time api
* Collectors
* forEach() in iterable interface

**2. What is a Lambda Expression, and why do we use them? Explain with a coding example and share the output screenshot.**

A **lambda expression** in Java is a short block of code that takes in parameters and returns a value.

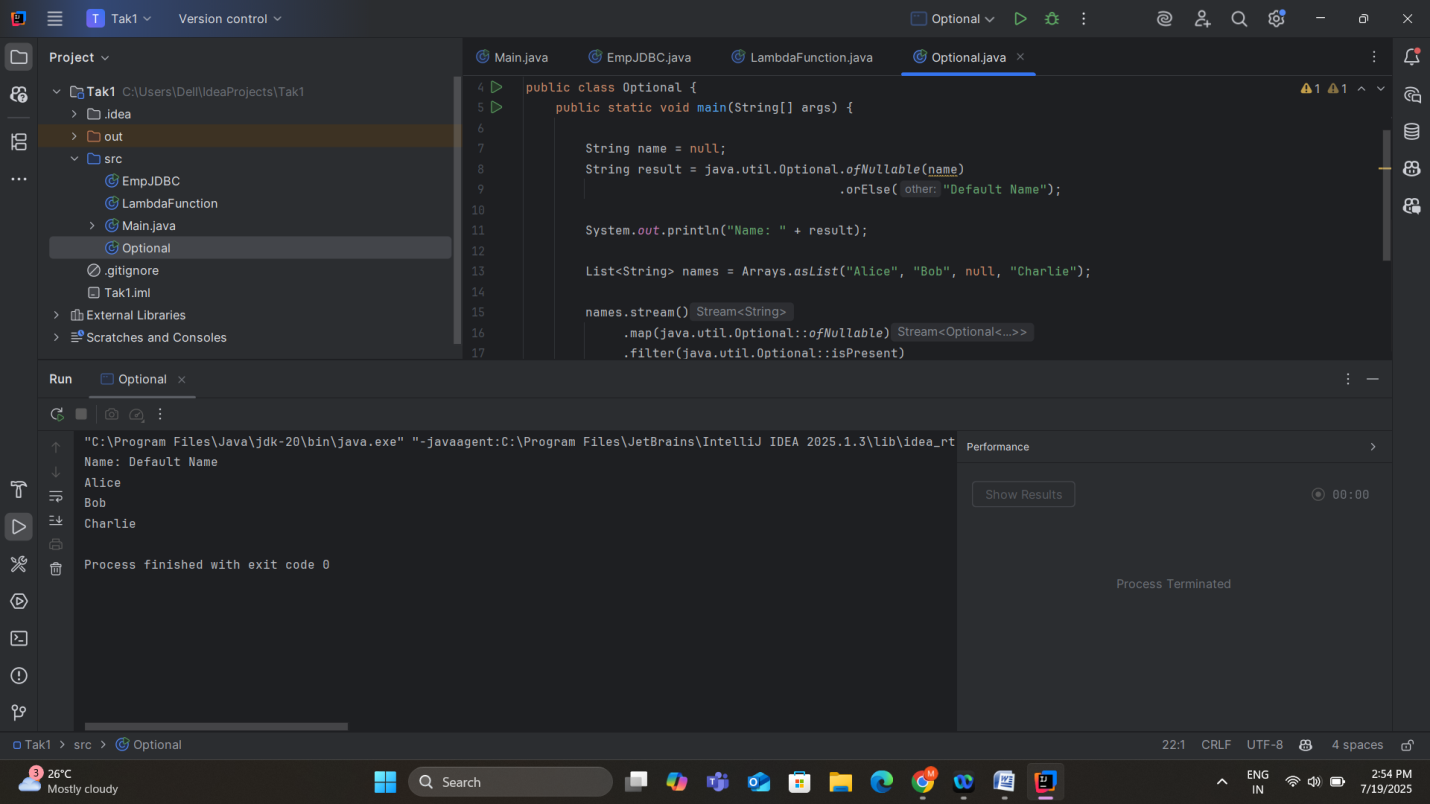
It makes code more readable, no need for anonymous class and oftern used with collections, streams.



**3. What is optional, and what is it best used for? Explain with a coding example and share the output screenshot.**

Optional<T> is a container object that is introduced in java 8 that may or may not contain the non-null value used to avoid the NullPointerException.

It reduces the risk of NullPointerException and provides the functions ifPresent(), orElse(), map().

****

4**. What is a functional interface? List some examples of predefined functional interfaces**

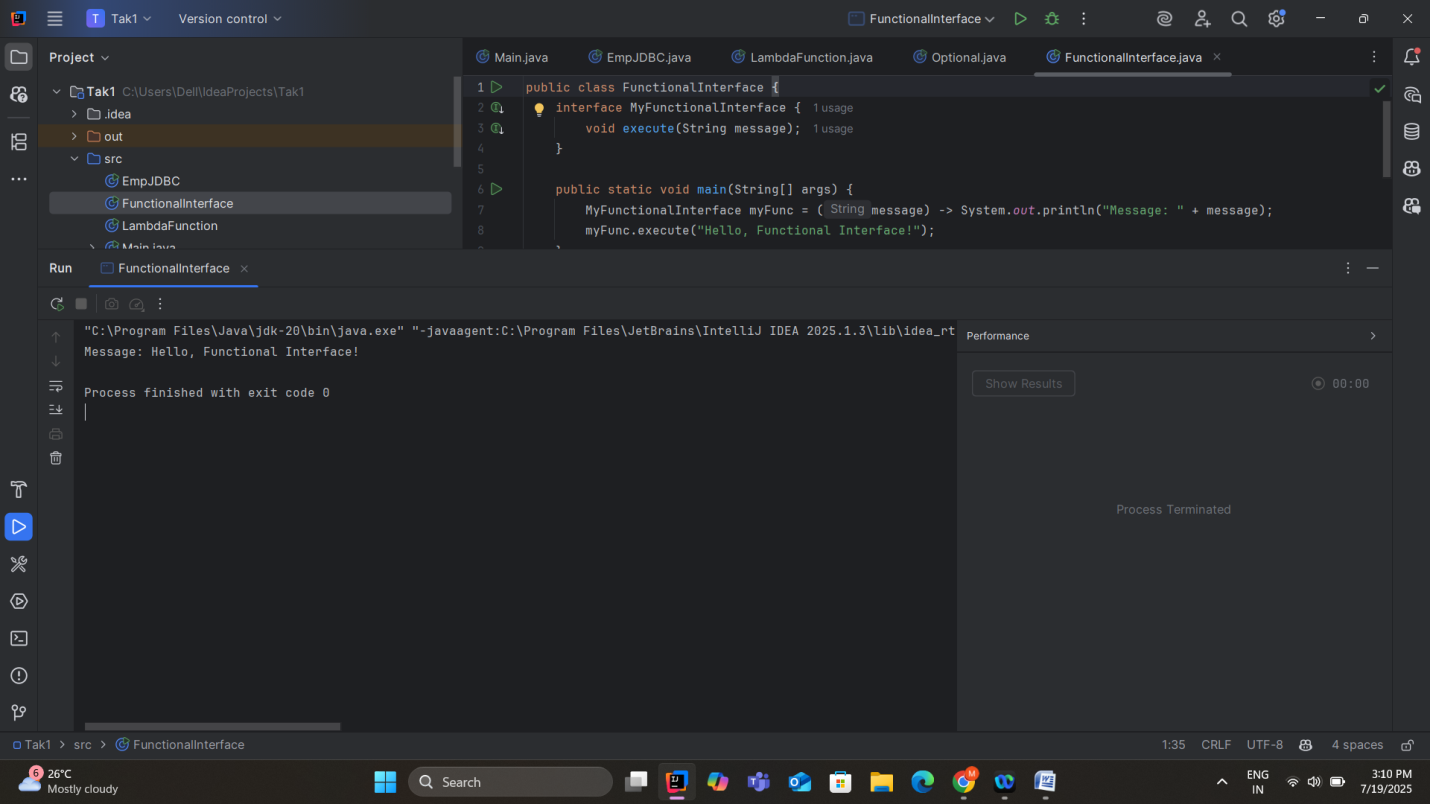
Functional Interface is an interface that contains exactly one abstract method. It can have multiple static or non static methods but only one abstract method.

Ex: Predicate<T> ---Interface  
 Boolean test(T,t)-----abstract method  
 Takes a value T and returns the Boolean.

Some of the predefined functional interfaces :  
 Function<T,R>  
 Consumer<T>  
 Supplier<T>

5**. How are functional interfaces and Lambda Expressions related?**

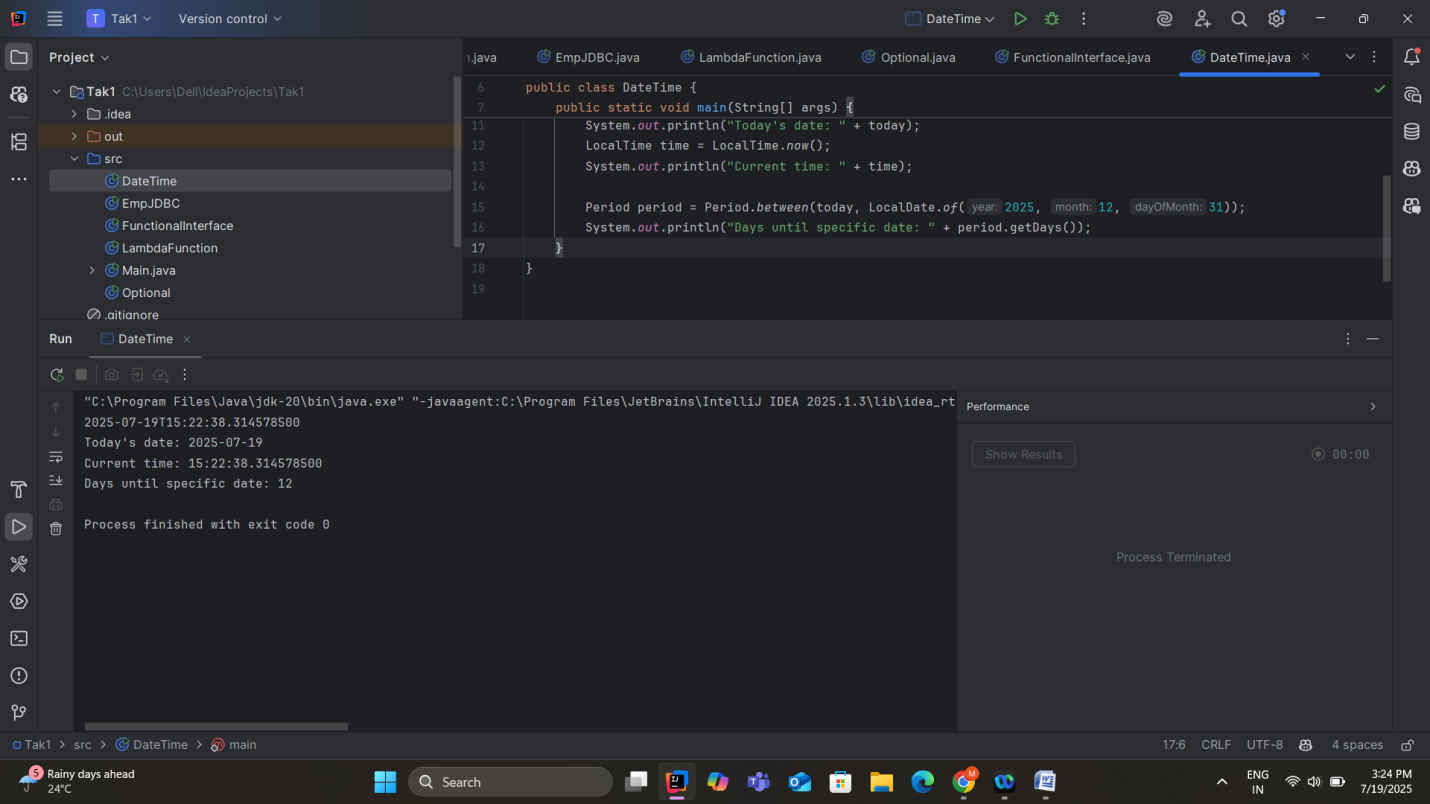
Lambda functions are used to implement the functional interfaces in java.



**6. List some Java 8 Date and Time API’s. How will you get the current date and time using Java 8 Date and Time API? Write the implementation and share the output screenshot.**

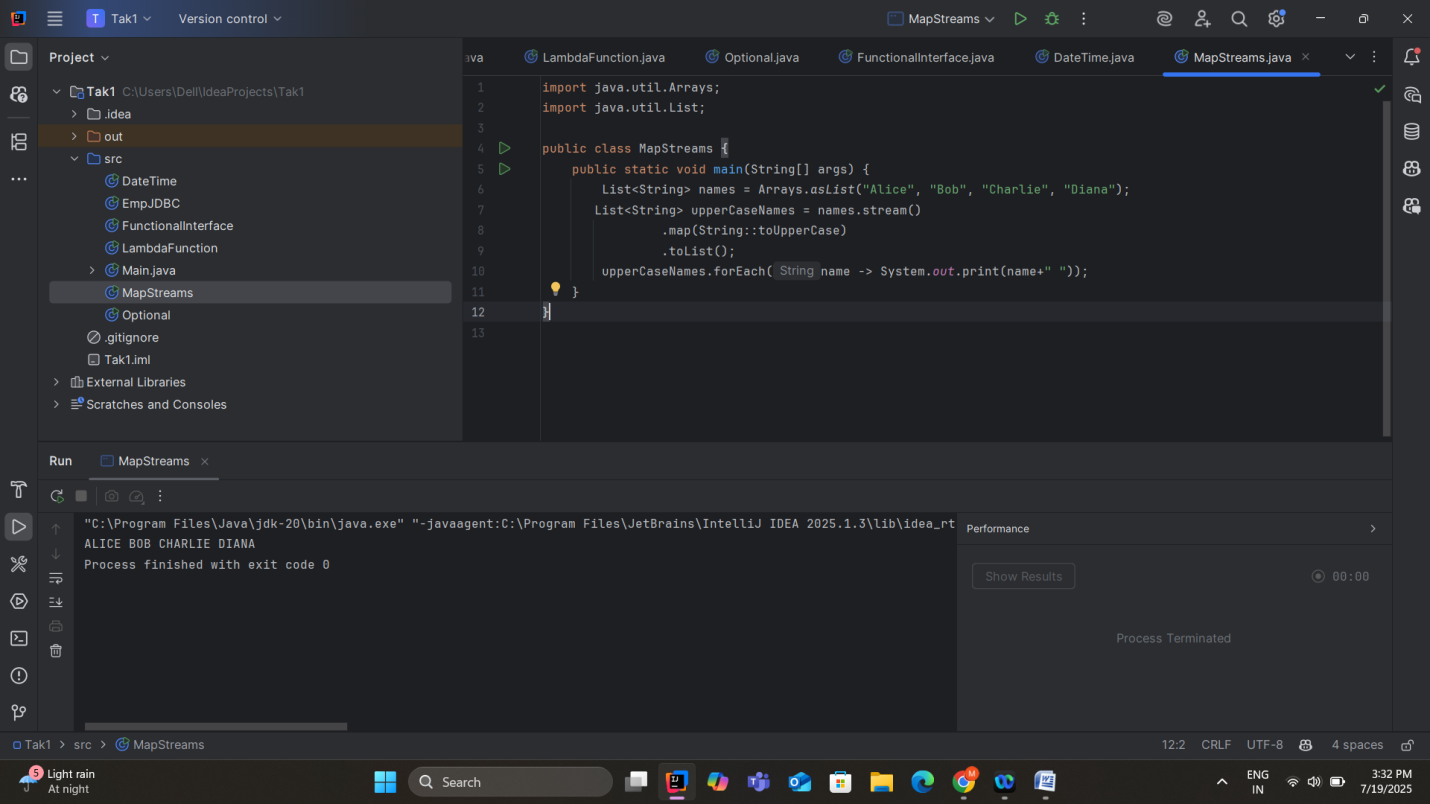
Key classes in java 8 Date & Time API:

* LocalDate
* LocalTime
* LocalDateTime
* ZonedTime
* Period
* Duration
* DateTimeFormatter



7. **How to use map to convert objects into Uppercase in Java 8? Write the implementation and share the output screenshot.**

We can use the map function in streams to convert the objects into uppercase

****

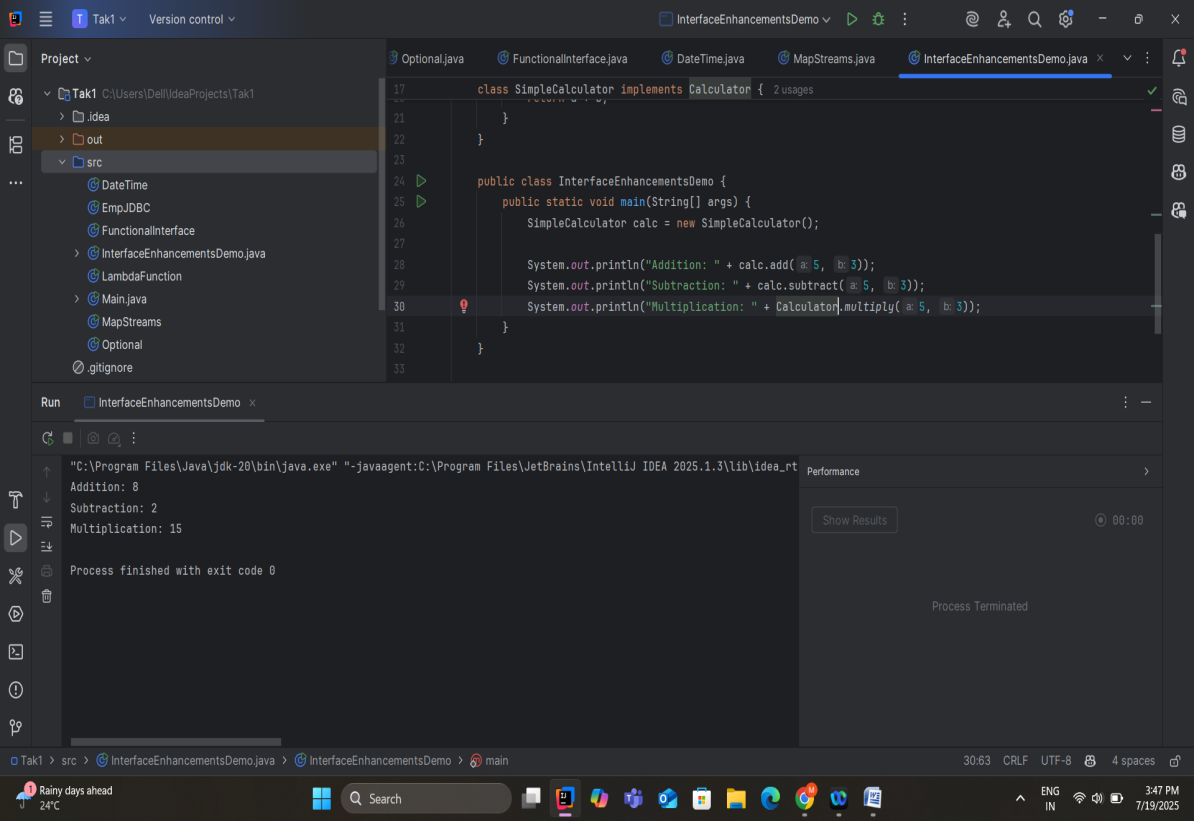
**8. Explain how Java 8 has enhanced interface functionality with default and static methods. Why were these features introduced, explain with a coding example?**

Before Java 8 interfaces could only implement the abstract methods without the method bodies. This made hard to evolve interfaces without breaking existing implementations.

Now with enhanced interface functionality the default and static methods are allowed inside the interface.

Why these features introduced:

* Evolve interfaces without breaking backward compatibility
* Group related utility methods inside the interface\



**9. Discuss the significance of the Stream API introduced in Java 8 for data processing. How does it improve application performance and developer productivity**

The **Stream API** is one of the most powerful and important features introduced in Java 8. It provides a **declarative, functional-style** approachto **process collections of data,** making yourcode **cleaner, more readable,** and **easier to parallelize.**

**Key Benefits of stream API:**

* Replaces complex for loops and if conditions with readable operations like filter(), map(), collect()
* Faster development
* Efficient data processing
* Support for parallelism
* Pipeline architecture

10. **What are method references in Java 8, and how do they complement the use of lambda expressions? Provide an example where a method reference is more suitable than a lambda expression. Explain with a coding example and share the output screenshot.**

Method references are the shorthand notation of lambda expressions to call a method by name. They make code more readable and concise especially when we are calling an existing method.

