**Date and Time formattes in java**

**Java.time.locaDateTime:**is the date where we see on calendar And (introduced in 1.8java)

**Java.util.date :** is an instant and actually represents a milllisecon

(introduced in 1.1java) many of its methods are deprecated

Returns **the number of milliseconds since January 1, 1970, 00:00:00 GMT** represented by this Date object.

**Java.time.calendar:**

(introduced in 1.1java) many of its methods are deprecated

1. **convertingtimetoAMPM():**

converting local date and time into String fomat in 12 hours format.

Using Date() and simpledateformat()

1. **datetimeformatteram():**

converting local date and time into String fomat in 12 hours format.

Using LocalDateTime and DateTimeFormatter.ofPattern("dd/MM/yyyy hh:mm a");

1. **convertDateToString("2022-11-27"):**

converting String date into object

using new SimpleDateFormat(date)

Get the date using calendar object

Convert the date into a

string using format() method

1. **convertingDateandTimetoString():**

converting given date and time into string format

using*java.time.format.DateTimeFormatter;*

1. **addingHoursTocurrentdateandtime():**

adding Hours To current date and time

using *java.util.Date;, java.util.Calendar;,java.text.SimpleDateFormat;*

1. **addingDaysTocurrentday():**

adding Hours To current date and time

using *java.util.Date;, java.util.Calendar;,java.text.SimpleDateFormat;*

1. **LocaldatetoinstantdateaddDays("2022-11-27",5):**

Converting localdate to instant date format and adding required hours into it for required results.

1. **LocaldatetoinstantdateaddingHours():**

Converting localdate to instant date format and adding required hours into it for required results.

1. **plushours():**

adding required hoursintolocaldate in simple way.

Using *java.time.LocalDateTime;*

1. **plusdays():**

adding required days into localdate in simple way.

Using *java.time.LocalDateTime;*

1. **countTheDifference("2022-12-20T15:22:22","2023-12-22T15:22:22"):**

calculating difference between two dates

using *java.time.temporal.ChronoUnit;*