**API’s & annotations Assignments by gaurav diwan**

**Java with DSA and System design**

Q1] program to display current date and time in java ?

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class DateTimeExample {

public static void main(String[] args) {

LocalDateTime currentDateTime = LocalDateTime.now();

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");

String formattedDateTime = currentDateTime.format(formatter);

System.out.println("Current Date and Time: " + formattedDateTime);

}

}

Q2] wap to convert a date to a string in the format “MM/DD/YYYY”.

import java.text.SimpleDateFormat;

import java.util.Date;

public class DateToStringExample {

public static void main(String[] args) {

Date currentDate = new Date();

SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy");

String formattedDate = sdf.format(currentDate);

System.out.println("Formatted Date: " + formattedDate);

}

}

Q3] what is the difference between collections and stream ? explain with an example.

Ans. **Collections:** Hold and manipulate data in memory. Eager evaluation.

**Stream:** Process data on-the-fly. Lazy evaluation.

// Collections

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

List<Integer> squares = new ArrayList<>();

for (Integer number : numbers) {

squares.add(number \* number);

}

// Stream

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

List<Integer> squares = numbers.stream()

.map(n -> n \* n)

.collect(Collectors.toList());

Q4] what is enums in java ? explain with an example.

Ans. **Enums in Java** are special data types used to define a fixed set of constants.

enum Day {

SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY

}

// Usage

Day today = Day.FRIDAY;

System.out.println("Today is " + today);

Q5] what are inbuilt annotations in java?

Ans. Some commonly used inbuilt annotations in Java include:

1. @Override: Indicates that a method overrides a method in its superclass.

2. @Deprecated: Marks a class, method, or field as deprecated, signaling that it should no longer be used.

3. @SuppressWarnings: Suppresses specific compiler warnings.

4. @FunctionalInterface: Declares an interface as a functional interface, allowing it to have only one abstract method.

5. @SafeVarargs: Indicates that the method does not perform potentially unsafe operations on its varargs parameter.

6. @Target: Specifies the types of program elements to which an annotation type can be applied.

7. @Retention: Specifies how long annotations with the annotated type are to be retained.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***