**Assignment- APIs and Annotation**

1. program to display current date and time in java.

Ans:

import java.time.\*;

public class DateTime {

public static void main(String[] args) {

LocalDate date = LocalDate.now();

System.out.println(date);

LocalTime time=LocalTime.now();

System.out.println(time);

}

}

Output:

javac DateTime.java

java DateTime

2023-03-30

10:33:11.025394800

2. Write a program to convert a date to a string in the format "MM/dd/yyyy".

Ans:

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

public class DateToString {

public static void main(String[] args) {

LocalDate date = LocalDate.of(2023, 4, 4);

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("MM/dd/yyyy");

String formattedDate = date.format(formatter);

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

}

}

Output:

javac DateToString.java

java DateToString

Formatted Date: 04/04/2023

3. What is the difference between collections and streams?Explain with an Example

Ans:

Collections and Streams are two different concepts in Java that are used to process and manipulate data.

A collection is a group of objects, such as a List, Set, or Map, that stores elements in memory. It allows for easy access and modification of the elements contained within it.

On the other hand, a stream is a sequence of elements that can be processed in a pipeline. Streams provide a functional programming approach to processing data, allowing for concise and efficient operations on large amounts of data.

The main difference between collections and streams is that collections are concrete data structures that store elements in memory, whereas streams are virtual views of data that can be processed on demand.

Ex: Collections

Output:

import java.io.\*;

import java.util.\*;

class Main {

public static void main(String[] args)

{

List<String> CompanyList = new ArrayList<>();

CompanyList.add("Google");

CompanyList.add("Apple");

CompanyList.add("Microsoft");

Comparator<String> com = (String o1, String o2) -> o1.compareTo(o2);

Collections.sort(CompanyList, com);

for (String name : CompanyList) {

System.out.println(name);

}

}

}

Ex: Streams

import java.io.\*;

import java.util.\*;

class Demo {

public static void main(String[] args)

{

List<String> CompanyList = new ArrayList<>();

CompanyList.add("Google");

CompanyList.add("Apple");

CompanyList.add("Microsoft");

CompanyList.stream().sorted().forEach(

System.out::println);

}

}

4. What is enums in java? explain with an example

Ans: We can use enum to define a group of named constants.

Enums are used to represent a collection of related constants that have a common purpose. Each constant in an enum is an instance of the enum type, and they are typically defined as public static final fields.

Here's an example of how to define an enum in Java:

class EnumDemo{

public enum DayOfWeek {

MONDAY,

TUESDAY,

WEDNESDAY,

THURSDAY,

FRIDAY,

SATURDAY,

SUNDAY

}

public static void main(String args[]){

for(DayOfWeek d:DayOfWeek.values())

System.out.println(d);

}

}

Here we define an enum called "DayOfWeek" that represents the days of the week. The enum has seven

constants, each representing a day of the week. The constants are defined in all uppercase letters by convention.

5. What are in built annotations in java

Ans: built-in annotations in Java:

@Override

@Deprecated

class EnumDemo{

public enum DayOfWeek {

MONDAY,

TUESDAY,

WEDNESDAY,

THURSDAY,

FRIDAY,

SATURDAY,

SUNDAY

}

public static void main(String args[]){

for(DayOfWeek d:DayOfWeek.values())

System.out.println(d);

}

}