

1) Program to display current date and time in java.

Ans:-

```
import java.time.*;
public class DateApi
{
    public static void main(String[] args) {
        java.time.LocalDate date = LocalDate.now();
        LocalTime time = LocalTime.now();
        System.out.println(date);
        System.out.println(time);
    }
}
```

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

Ans:-

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
public class ConvertDate {
    public static void main(String[] args) {
        LocalDate date = LocalDate.now();
        DateTimeFormatter formatter =
            DateTimeFormatter.ofPattern("MM/dd/yyyy");
        String formattedDate = date.format(formatter);

        System.out.println("Formmated Date: "+formattedDate);
    }
}
```

3) What is the difference between collections and stream? Explain with an example.

Ans:-

stream	collections
When we want to perform some operations on collection without affecting its data then we should make separate stream on collection data and then we perform some operation on that stream.	When we want to represent a group of individual objects as a single entity then we should use collections.

Stream are consumable, Cannot perform more operation on a stream	Collections are not consumable Can perform many operation on a collection without creating it again
Streams are not modifiable	Collections are modifiable
Streams are iterated internally by just mentioning operations.	Collections are iterated using external loops.
Use functional interface which enable us to use lambda expression.	Collection does not use functional interface

Collections example:

```
import java.io.*;
import java.util.*;
public class CollectionsDemo {
    public static void main(String[] args) {

        ArrayList<String> al = new ArrayList<>();
        al.add("usama");
        al.add("ayesha");
        al.add("talha");

        Comparator<String> com = (String i1, String i2) -> i1.compareTo(i2);
        // ascending order
        Collections.sort(al); // sort method of Collection utility class
        which method can use with any Collection

        for(String name : al)
        {
            System.out.println(name);
        }
    }
}
```

Stream example:

```
package Naresh;
import java.util.*;
```

```

public class StreamDemo {
    public static void main(String[] args) {
        List<Integer> list = Arrays.asList(2,4,6,8);

        list.stream().sorted().forEach(n->System.out.println(n));
    }
}

```

4) What is enums in java? Explain with an example.

Ans:- An enums is a special class that represents a group of constants(unchangeable variables like final variables).

An enum type is a special data type that enables for a variable to be a set of predefined constants. Common example include compass direction(EAST, WEST, NORTH, SOUTH) and the days of the week.

Example:-

Class EnumDemo

```

{
    Public static void main(String[] argos)
    {
        Enum Level
        {
            LOW,
            MEDIUM,
            HIGH
        }

        for(Level myVar : Level.values())
        {
            System.out.println(myVar);
        }
    }
}

```

5) What are inbuilt annotation in java?

Ans:- @Override

@Deprecated

@SafervarArgs

@ SuppressWarnings

@FunctionalInterface

@Target

@Inherited
@Retention