List of Concepts Involved:

- Date and Time API in java
- Stream API in Java
- Enums
- What is Annotation?
- In Built Annotation
- Custom Annotation
- Reflection API in Java

Date and Time API in java

Date and Time API: (Joda-Time API)
Until Java 1.7 version the classes present in Java.util package to handle Date and Time (like Date, Calendar, TimeZone etc) are not up to the mark with respect to convenience and performance.

To overcome this problem in the 1.8 version oracle people introduced Joda-Time API.
This API developed by joda.org and available in Java in the form of the "java.time" package.

Ex: program to display System Date and time.

```
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
2022-10-30
09:13:34.692

java.util.Date vs java.sql.Date

```
public class Test {
 public static void main(String[] args) {
  java.util.Date utilDate = new java.util.Date();
  System.out.println(utilDate);

  long l = utilDate.getTime();

  java.sql.Date sqlDate = new java.sql.Date(l);
  System.out.println(sqlDate);
 }
}
```

Output
Sun Oct 30 10:05:33 IST 2022
2022-10-30

Difference b/w java.util.Date and java.sql.Date

java.util.Date

- It is a utility class to handle Date in our java program.
- It represents both Date and Time

java.sql.Date

- It is designed class to handle Dates w.r.t DB operations
- It represents only Date,but not Time.

Note: In sql package

Time(C) => Time value
TimeStamp(C) => Date and Time value

Stream API in Java

Streams

To process objects of the collection, in 1.8 version Streams concept introduced.

What is the difference between Java.util.streams and Java.io streams?

java.util streams meant for processing objects from the collection. Ie, it represents a stream of objects from the collection but Java.io streams are meant for processing binary and character data with respect to file. i.e it represents a stream of binary data or character data from the file. Hence Java.io streams and Java.util streams both are different.

What is the difference between collection and stream?

- If we want to represent a group of individual objects as a single entity then We should go for collection.
- If we want to process a group of objects from the collection then we should go for streams.
- We can create a stream object to the collection by using the stream() method of the Collection interface.
- stream() method is a default method added to the Collection in 1.8 version.

```
default Stream stream()
Ex: Stream s = c.stream();
import java.util.*;
import java.util.stream.*;

public class Test {
 public static void main(String[] args) {
   ArrayList<Integer> al = new ArrayList<Integer>();
   al.add(0);
   al.add(5);
   al.add(10);
   al.add(15);
   al.add(20);
   al.add(25);

   System.out.println(al);
```