

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(c
            -> System.out.println(c));
    }
}
```

o/p:- Apple
Google
Microsoft.

Q4) what is enums in java? Explain with an example.

Ans:- we can use enum to define a group of named constants. Enum 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.

5
Ans:-

Q
Q
Q
Ja
Ja
bro

Ques: What is Enum?

Ans: Enum is a class-like type that represents a set of constant values.

Monday,

Tuesday,

Wednesday,

Thursday,

Friday,

Saturday,

Sunday.

}

public static void main (String[] args) {

{

for (Day of week d : Day of week.values())
System.out.println(d);

}

}

Here we define an enum called "Day of week". that represents the day of the week. The enum has seven constants, each representing a day of the week. The constants are defined in all uppercase letter by convention.

Ques: What are built-in annotations in Java?

Ans:-

built-in annotations in Java.

@Override @SuppressWarnings @FunctionalInterface

@Deprecated @Retention @Target @Documented

@Inherited. These built-in annotations in

Java are used to provide additional information to the Java compiler and other tools. They help improve code readability, maintainability and safety by enforcing specific rules and behaviors in Java code.

- ④ Java Streams support both sequential and parallel processing.
- ④ It supports parallel processing and parallel processing can be very helpful in achieving high performance.
- ⑤ streams are iterated internally by just mentioning the operations.
- ⑤ collections are iterated externally using loops.

Ex. Collections :

```
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);
```

```
        }
```

O/p
Apple
Google
Microsoft

String formattedDate = date.format(formatter);

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

O/p

Java Date To String.java

Java Date To String

Formatted Date : 04/04/2023

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

Ans:

STREAMS

COLLECTIONS

- ① It doesn't store data. It operates on the source data structure i.e. collection currently has in a particular data structure like set, list or map.
- ② They use functional interface like lambda which makes it a good fit for programming language.
- ③ Java streams are consumable i.e. to traverse the stream it needs to be created every time.
- ④ They don't use functional interface.
- ⑤ They are non-consumable i.e. can be traversable multiple times without creating it again.

Important APIs and Annotation

Page No. 28
Date / /

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

Ans: `import java.time.*;`

```
public class Datetime {  
    public static void main (String[] args) {  
        LocalDate date = Local LocalDate.now();  
        System.out.println (date);  
        LocalDateTime time = LocalDateTime.now();  
        System.out.println (time);  
    }  
}
```

O/P:-

```
javac Datetime.java
```

```
java Date.time
```

```
2023-03-30
```

```
2023-03-30T11:02:53.448000
```

Q2) write 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 DateToString {  
    public static void main (String[] args) {  
        LocalDate date = LocalDate.of (2023, 4, 23);  
        DateTimeFormatter formatter = DateTimeFormatter  
            ofPattern ("mm/dd/yyyy");  
    }  
}
```