3rd April 23 Important APIs and Annotation Assignment - 25

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);
}
}
```

3rd April 23 Important APIs and Annotation Assignment - 25

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: STREAMS

- It doesn't store data, it operates on the source data structure i.e collection.
- They use functional interfaces like lambda which makes it a good fit for programming languages.
- Java Streams are consumable i.e; to traverse the steam, it needs to be created every time.
- Java streams support both sequential and parallel processing.
- Streams are iterated internally by just mentioning the operations.

COLLECTIONS

- It stores/holds all the data that the data structure currently has in a particular data structure like **Set, List** or **Map**.
- They don't use functional interfaces.
- They are non-consumable i.e; can be traversable multiple times without creating it again.
- It supports parallel processing and parallel processing can be very helpful in achieving high performance.
- 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);
}
}
}
```

Output:

Apple
Google
Microsoft

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);
}
}
```

Output:

Apple
Google
Microsoft

3rd April 23 Important APIs and Annotation Assignment - 25

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
@SuppressWarnings
@FunctionalInterface
@Retention

3rd April 23 Important APIs and Annotation Assignment - 25

@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.