APIs and Annotation

Assignment Solution

Q.1 program to display current date and time in Java?

Ans.1
```
        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
 2024-07-02
 10:38:11.025394800
```

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

Ans.2
```
    import java.time.LocalDate;
  import java.time.format.DateTimeFormatter;
  public class DateToString{
      public static void main(String[] args) {
```

```
LocalDate date = LocalDate.of(2024, ~~7~~, 2);
  DateTimeFormatter formatter =
     DateTimeFormatter.ofpattern("MM/dd/y
  String formattedDate = date.format(formatter)

   System.out.println("Formatted Date: " + forma
                                     -tedDate);
  }
}
```

output:

```
javac DateToString.java
java DateToString.
Formatted Date: 07/02/2024.
```

Q.3 What is the difference b/w Collections and Streams? Explain with an Example.

Ans.3

Streams	Collections
1. It doesn't store date, it operates on the source data structure i.e Collection.	1. It stores/hold all the data that the data structure currently has in a particular data structure like Set, List or Map.
2. They uses functional interfaces like Lambda which makes it a good fit for programming language	2. they don't use functional interfaces

YY");

3. Streams are iterated internally by just mentioning the operations.	3. 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 Array
      CompanyList.add("Google");
      CompanyList.add("Microsoft");
      CompanyList.stream().sorted().forEa
                System.out::println);
    }
}
```

output:
```
Apple
Google
Microsoft
```

Q4 what is enums in Java? Explain with example

Ans4 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 "Dayof-week" that represents the day of week. the enum has seven constants, each representing a day of the week. the constants are defined in all uppercase letter as convention.

Q.5 What are in built annotations in Java.

Ans's built-in annotations in Java.

@override
@Deprecated
@ SuppressWarnings
@ Relation
@functionalInterface
@ 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 saftey by enforcing specific rules and behaviours in Java Code.