**Interview Questions: Set 554**

1. What are the different new classes introduced in Date API?

LocalDate, LocalTime, LocalDateTime, Calender

1. How do you format the current date?

LocalDateTime lt = LocalDateTime.now();

System.out.println(lt);

// 2022-09-26T20:15:03.917767500

DateTimeFormatter df = DateTimeFormatter.ofPattern("dd-MMM-yyyy H:ma");

String output = lt.format(df);

System.out.println(output);`

//26-Sep-2022 20:15pm

1. What is the difference between LocalDate, LocalTime, LocalDateTime ?

We can print only date with the help of LocalDate, we can print only time with the help of LocalTime and we can print date and time both with the help of LocalDateTime

LocalDate has plusDays() and minusDays() methods, while LocalTime has a plusMinutes() and minusMinutes() methods,LocatDateTime has plusYears() and minusYears() method

LocalDate — A date without a time-zone.

LocalTime — A time without a time-zone.

LocalDateTime — A date-time without a time-zone.

ZonedDateTime — A date-time with a time-zone.

Duration — A time-based amount of time.

Period — A date-based amount of time.

1. What are the different ways to create the LocalDate instance?

Using now() method

LocalDate date = LocalDate.now();

LocalDate yesterday = date.minusDays(1);

LocalDate tomorrow = yesterday.plusDays(2);

Using of() method

LocalDate date1 = LocalDate.of(2017, 1, 13);

System.out.println(date1.isLeapYear());

LocalDate date2 = LocalDate.of(2016, 9, 23);

System.out.println(date2.isLeapYear());

1. What is the use of Duration and the Period Class ChronoUnit?

we can use Period, Duration or ChronoUnit to calculate the difference between two LocalDate or LocaldateTime . Period to calculate the difference between two LocalDate. Duration to calculate the difference between two LocalDateTime . ChronoUnit for everything.

1. How do you get the difference of dates between 2 dates?

Using getDay() method we can find the diff of dates bw 2 days

LocalDate from = LocalDate.of(2021, 9, 6);

LocalDate to = LocalDate.of(2022, 9, 26);

Period period = Period.between(from, to);

int diffofdate = Period.between(from, to).getDays();

System.out.println("date diff is: " + diffofdate);

1. What is the difference between FileReader and FileInputStream?

Entire fileio divided into 2 file hierarchy

**1.Character Hierarchy**

**FileWriter:** (means : if i want open the file into write mode then we use filewriter this is come under character hierarchy)

try {

FileWriter fw = new FileWriter(new File("fofile.txt"), true); //true means append mode

String str = "hello mr akash, ";

fw.write(str);

System.out.println("file is created");

fw.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

**FileReader:** (means : if i want open the file into read mode then we use filereader this is come under character hierarchy)

try {

FileReader fd = new FileReader(new File("fofile.txt"));

int c;

while ((c = fd.read()) != -1) {

System.out.print((char) c);

}

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

**2. Byte Hierarchy**

**FileOutputStream:** if i want to write into file using byte hierarchy (means: if i want open the file into write mode)

try {

// opening file into write/append mode mode

FileOutputStream fo = new FileOutputStream("first.txt", true);

byte[] data = "we are learning java, and spring".getBytes();

fo.write(data);

System.out.println("data added into file");

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

**FileInputStream:** if i want to read from file using byte hierarchy (means: if i want to open the file into read mode)

try {

// opening the file into read mode

FileInputStream fes = new FileInputStream(new File("first.txt"));

int i = 0;

while ((i = fes.read()) != -1) {

System.out.print((char) i + "");

}

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

1. What is serialization? And how do you achieve it?

**Serialization:-- Storing the object state in the file.**

Steps: Serialization

1. Create a class that implemnets serializable (marker interface)

2. Create object of that class

3. Open any file in the write mode, using FileOutputStream

4. Pas the fileOuputStream to the OBjectOutputStream

5. Invoke the writeObject() method of objectoutputstream, and pass the object that you

want to serialize.

**Deserialization: -- Retriving the object state from file**

Steps: Deserialization

1. Open the file where the objects are serialized in the read mode.

using FileInputStream

2. Pass FileInputStream to OBjectInputStream

3. Invoke the readObject() method of objectinputstream, and cast the

object type (e.g. employee) to the object returned.

1. What is the Externizable interface?
2. What is the use of defaultWriteObject() and defaultReadObject()?

**defaultWriteObject()**

**Serialize the object in default way this way i am explicitly storing the information of date**

Write the non-static and non-transient fields of the current class to this stream. This may only be called from the writeObject method of the class being serialized. It will throw the NotActiveException if it is called otherwise

**defaultReadObject()**

Read the non-static and non-transient fields of the current class from this stream. This may only be called from the readObject method of the class being deserialized. It will throw the NotActiveException if it is called otherwise

1. What is the use of transient keyword?

If i want particular element is not serialize then i make it as a transient

Ex: transient int roll no; // transient variable are not part of serializable.

**Why we use ransient keyword: if i write an account no or password that should not be**  **part of serialize means for security purpose**

1. How do you achieve the serialization in containment, if the contained object is not serializable?
2. How does the serialization work in the inheritance scenario?
3. Which method is present in Serializable interface?

Serializable interface is a marker interface there is no method present inside it.

1. When do you override readObject and writeObject method? From which class they are overridden?
2. What is the use of BufferedReader object?
3. Do you know anything about NIO?
4. How do you work with Random Access File?