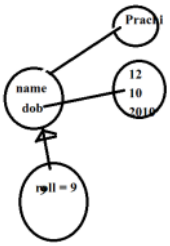
Monday, November 22, 2021 10:33 AM

File Storing **only property values** of object in the file Prachi121020109	Serialization technique of saving the object ---------- Property values + METADATA Study.Person.name=prachi, Study.Person.MyDate.dob.day = 12,study.Person.MyDate. dob.month=10, study.Person.MyDate.dob.year=2010 , study.Student isa Person study.Student.roll = 9	DeSerialization --- Reading the serailized file and creating object structure and populating the properties in RAM 
Problem --- Even though data is saved but it cannot be read back properly	Advantage of serialization - we can recreate the same object structure as it is while reading the file again	

Scenarios ----------

1	If Student is a Person Only Student is Serializable and Person is not Serializable	Student properties are saved in file but person properties are not saved. Person will get default property values during Deserialization ---- Person [name=null, dob=null] Student [roll=9]
2	If Person and Student are Serializable but Person Has MyDate3 and MyDate3 is not Serializable	Exception occurs during serialization java.io.NotSerializableException: study.collections.MyDate3 NOTHING IS SAVED
3	If Person, Student , MyDate3 all are Serializable but roll number is TRANSIENT	All other properties will be SAVED except the transient property roll number !!! Roll number is not serialized because it is transient. During deserialization roll number gets default value/ value in default constructor

---

```
Stream   CLASS
        Stream filter(PredicateInterface   obj)
              {
                    Newstream
                          For(…length of list……..)
                            if(obj.test(listelement))
                                 Newstream(listelement)

                    Return newstream
              }
        void  foreach (ConsumerInterface obj)
       Stream map(SomeInterface obj)
```

```
list  = Arrays.asList(13,4,5,6,7,8,9,10,13,35,,78,89,435,666,777,888 )
Stream streamObj  =        list .stream();

        streamObj.filter()

        streamObj.foreach()

         streamObj.map()
```

```
          streamObj.filter(  ????    )
                        . Map()
                        .foreach()
```

Ways to create object of Predicate Interface ???

1.

```
 class XYZ  implements PredicateInterface
 {
                 boolean test(String element )
                 {
                        If ( element.length < 20 ) return true ; else return false
                 }
 }
    streamobj . Filter ( new XYZ() )
```

2.
```
   PredicateInterface   obj  = new PredicateInterface() {

                        boolean test(String element )
                        {
                              If ( element.length < 20 ) return true ; else return false
                        }

                  }
        Streamobj . Filter ( obj )
```

3.
```
    PredicateInterface   obj = (element)->{  If ( element.length < 20 ) return true ; else return false}  ;

                        Streamobj . Filter ( obj )
```

4.
```
      streamobj . Filter (  (element)->{  If ( element.length < 20 ) return true ; else return false}   );
```

MultiThreading ----------

Process = Is a executing program

Java process Space = JVM = Class Area, Stack Area, Heap Area

Thread = path of execution within a process

Thread Life Cycle / Process Life Cycle

Thread creation / process creation

Multithreading examples

Data Sharing in Thread --- Account

Problem ---- RACE CONDITION

SOLUTION ----- MUTUAL EXCLUSION / SEMAPHORES

---

1 main

Many threads !!!

Main - has the path of execution of DEFAULT Thread

java .lang. Thread class !!!

Public void run () ---- this method has the path of execution of threads

So when we want to create new threads ---- we can override public void run ()

TWO ways to create threads in Java

1. overriding run method in Thread class
   Extedns Thread

2. overriding run method in Runnable interface
   Implements Runnable

THUMB RULE ----------

Lambda should be used when the class is needed only **once** and the definition of the interface method is **short** (2 ,3 lines) !!

If you are using the class at many places -then use named class
If the definition of the interface method is long then go for named class!!!

Thread API

Run	We give the path of execution of the thread here
Start	To add the thread at OS level to ready Queue
setName / getName	
currentThread	Static method of Thread class that returns the reference of current thread
sleep()	Static method

HW ----------

0. Try Serializable code done in class

1. Write 2 threads using extends Thread
   a. Thread1 calculates the square of each number starting from 1 , displays the square and sleeps for 2sec it continues till 50
   b. Thread 2 calculates the factorial of of each number starting from 1 , displays the square and sleeps for 2sec it continues till 15

   Create threads in main, give name to each thread print the name

2. Write 2 threads using Runnable
   a. Thread1 prints the table of each number starting from 1 , displays the square and sleeps for 2sec it continues till 50 in file tables.txt (PrintWriter)
   b. Thread 2 accepts a number from the user and shows the star triangle
      User enters 3
      1
      1 2
      1 2 3

      do this for 5 times .
      Give a name to each thread and print it
