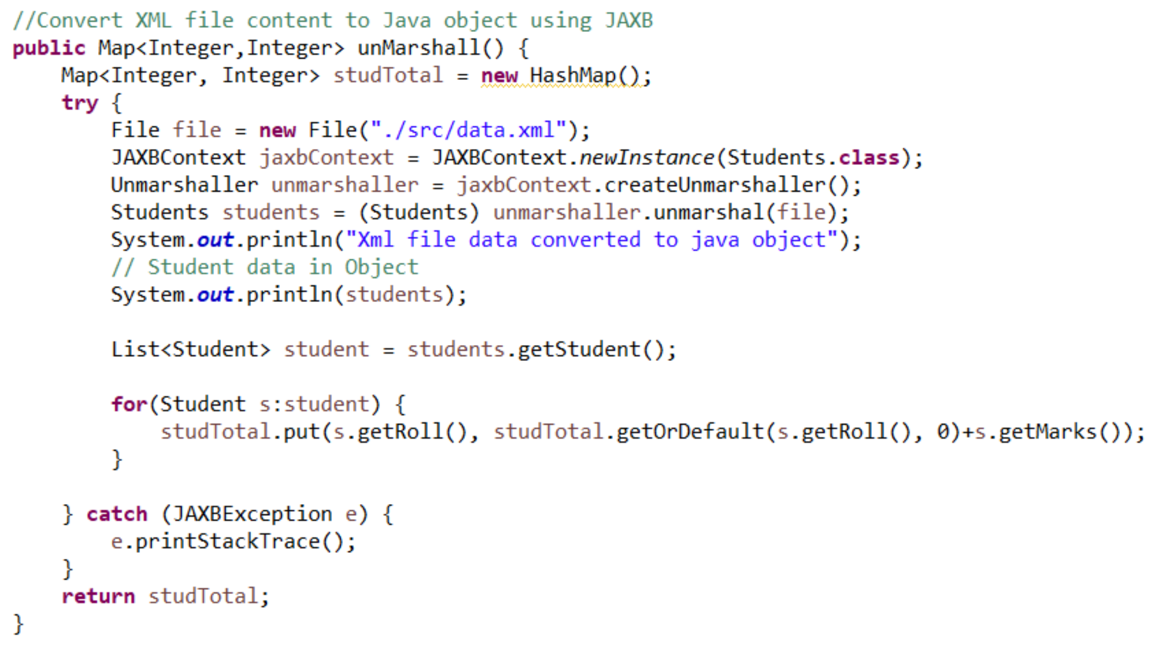
Spring AU ’21 – Java Fundamentals – Morning Session

Name: Sheik Abudhahir K

Date: 18/01/2021

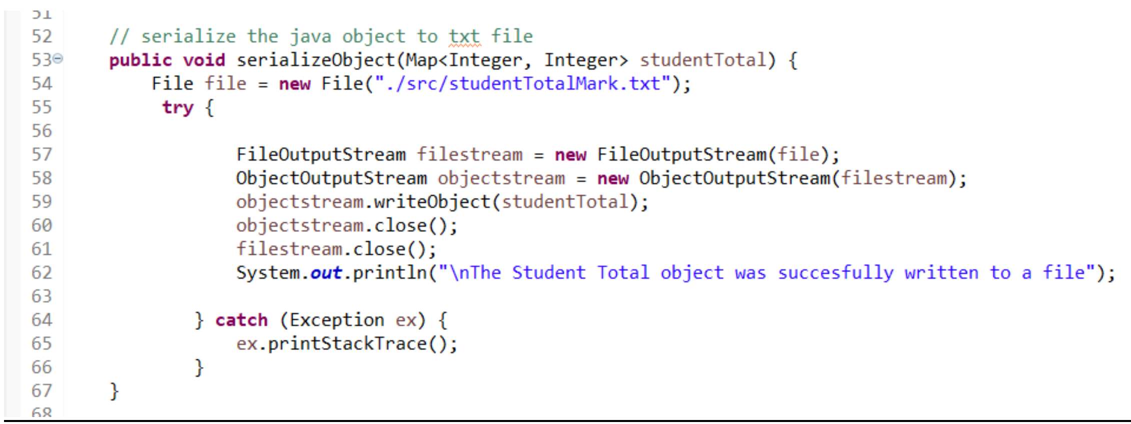
Source code is uploaded

Read an XML file “”student.xml” containing list of student data in the following format, deserialize them into java objects



```
//Convert XML file content to Java object using JAXB
public Map<Integer,Integer> unMarshall() {
    Map<Integer, Integer> studTotal = new HashMap();
    try {
        File file = new File("./src/data.xml");
        JAXBContext jaxbContext = JAXBContext.newInstance(Students.class);
        Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
        Students students = (Students) unmarshaller.unmarshal(file);
        System.out.println("Xml file data converted to java object");
        // Student data in Object
        System.out.println(students);

        List<Student> student = students.getStudent();

        for(Student s:student) {
            studTotal.put(s.getRoll(), studTotal.getOrDefault(s.getRoll(), 0)+s.getMarks());
        }

    } catch (JAXBException e) {
        e.printStackTrace();
    }
    return studTotal;
}
```

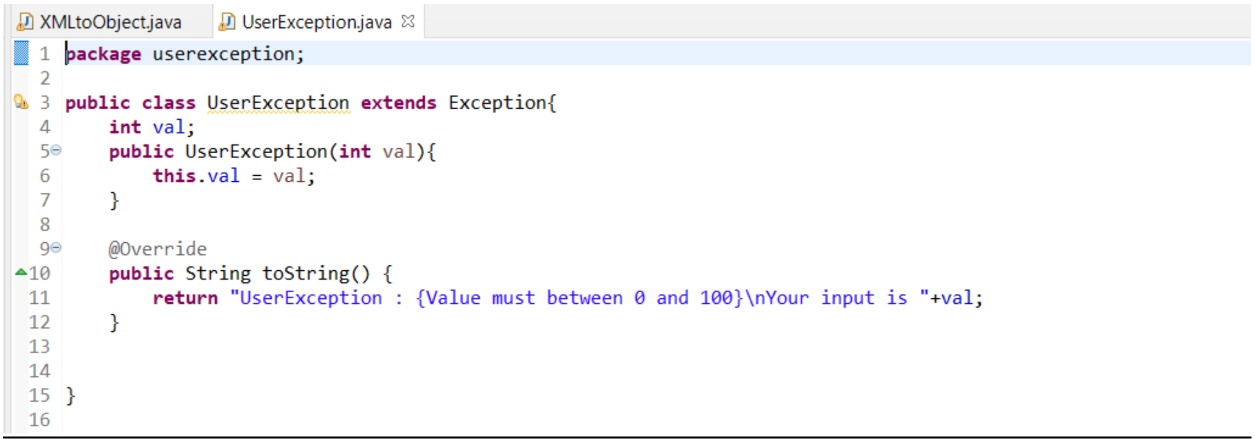
then serialize the unique <rollnumber, total marks(Phy+chem+math)> to a text file “student.txt” with appropriate exception handling.



```
    // serialize the java object to txt file
    public void serializeObject(Map<Integer, Integer> studentTotal) {
        File file = new File("./src/studentTotalMark.txt");
         try {

                FileOutputStream filestream = new FileOutputStream(file);
                ObjectOutputStream objectstream = new ObjectOutputStream(filestream);
                objectstream.writeObject(studentTotal);
                objectstream.close();
                filestream.close();
                System.out.println("\nThe Student Total object was succesfully written to a file");

            } catch (Exception ex) {
                ex.printStackTrace();
            }
    }
```

Here I use map to store the unique<rollnumber, totalmarks>, serialize the map object to text file

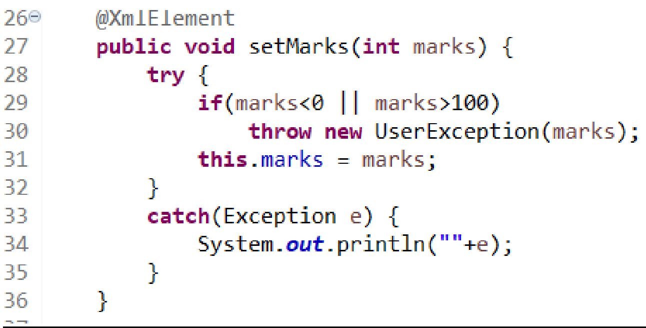
Handle the exception for marks, if enter marks in negative or marks above 100 then it throws exception



XMLtoObject.java UserException.java

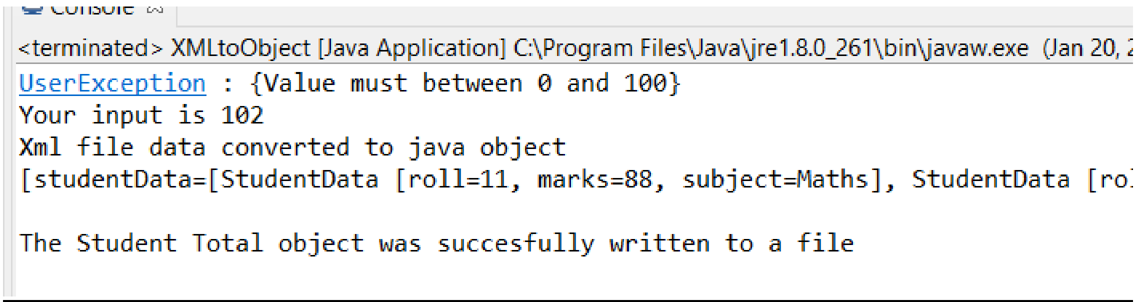
```java
package userexception;

public class UserException extends Exception{
    int val;
    public UserException(int val){
        this.val = val;
    }

    @Override
    public String toString() {
        return "UserException : {Value must between 0 and 100}\nYour input is "+val;
    }


}

```

Used in student.java



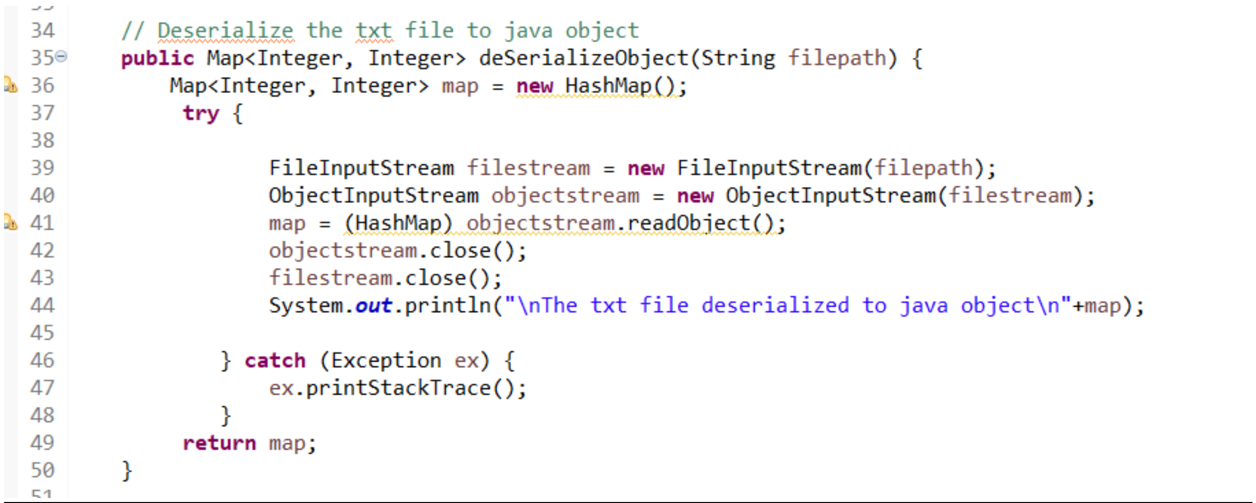
```java
    @XmlElement
    public void setMarks(int marks) {
        try {
            if(marks<0 || marks>100)
                throw new UserException(marks);
            this.marks = marks;
        }
        catch(Exception e) {
            System.out.println(""+e);
        }
    }
```

Output:



```
<terminated> XMLtoObject [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe  (Jan 20, 2
UserException : {Value must between 0 and 100}
Your input is 102
Xml file data converted to java object
[studentData=[StudentData [roll=11, marks=88, subject=Maths], StudentData [ro

The Student Total object was succesfully written to a file
```

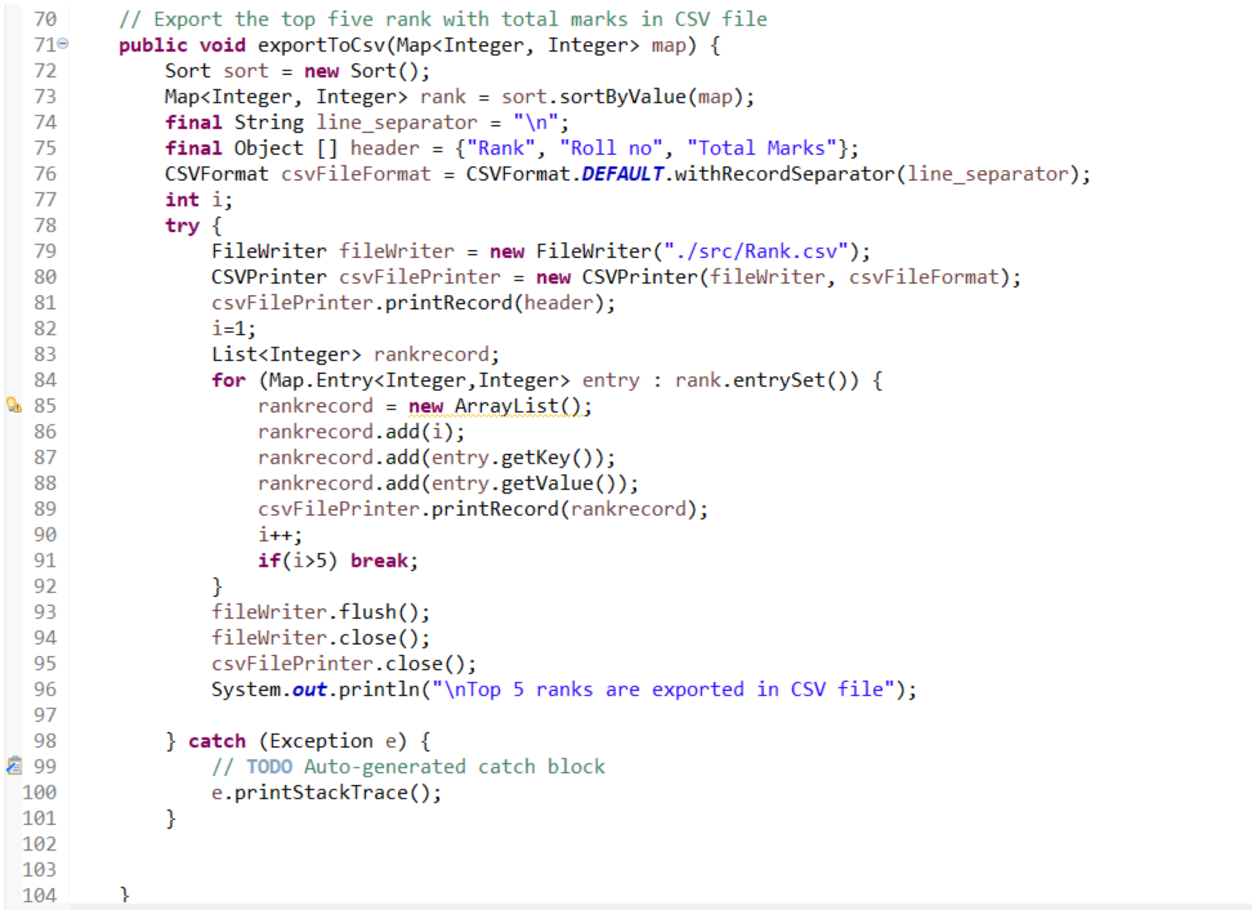
Secondly, deserialize the previously stored student.txt file with roll num and total marks.



```
// Deserialize the txt file to java object
public Map<Integer, Integer> deSerializeObject(String filepath) {
    Map<Integer, Integer> map = new HashMap();
     try {

            FileInputStream filestream = new FileInputStream(filepath);
            ObjectInputStream objectstream = new ObjectInputStream(filestream);
            map = (HashMap) objectstream.readObject();
            objectstream.close();
            filestream.close();
            System.out.println("\nThe txt file deserialized to java object\n"+map);

        } catch (Exception ex) {
            ex.printStackTrace();
        }
     return map;
}
```

Here I deserialize the text file and get the rollno and total marks then stored again in another map

serialize the top 5 students' roll numbers and total marks into excel or csv file in the following format.



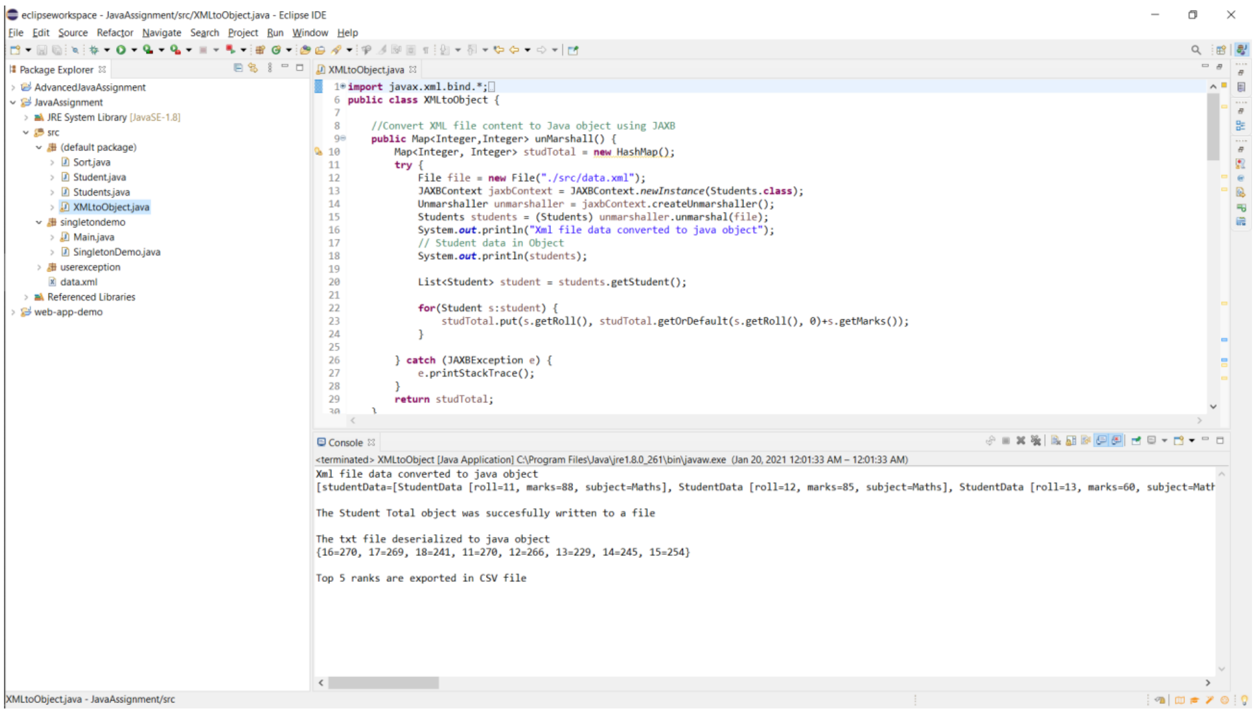
```
// Export the top five rank with total marks in CSV file
public void exportToCsv(Map<Integer, Integer> map) {
    Sort sort = new Sort();
    Map<Integer, Integer> rank = sort.sortByValue(map);
    final String line_separator = "\n";
    final Object [] header = {"Rank", "Roll no", "Total Marks"};
    CSVFormat csvFileFormat = CSVFormat.DEFAULT.withRecordSeparator(line_separator);
    int i;
    try {
        FileWriter fileWriter = new FileWriter("./src/Rank.csv");
        CSVPrinter csvFilePrinter = new CSVPrinter(fileWriter, csvFileFormat);
        csvFilePrinter.printRecord(header);
        i=1;
        List<Integer> rankrecord;
        for (Map.Entry<Integer,Integer> entry : rank.entrySet()) {
            rankrecord = new ArrayList();
            rankrecord.add(i);
            rankrecord.add(entry.getKey());
            rankrecord.add(entry.getValue());
            csvFilePrinter.printRecord(rankrecord);
            i++;
            if(i>5) break;
        }
        fileWriter.flush();
        fileWriter.close();
        csvFilePrinter.close();
        System.out.println("\nTop 5 ranks are exported in CSV file");

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }


}
```

Here I write one method for sort by total marks for ranking and store the top5 data in csv

Main method:

```java
public static void main(String[] args) {
    // TODO Auto-generated method stub
    XMLtoObject object = new XMLtoObject();
    Map<Integer, Integer> studentTotal = new HashMap();
    studentTotal = object.unMarshall();

    object.serializeObject(studentTotal);

    Map<Integer, Integer> deSerializeObject = new HashMap();
    deSerializeObject = object.deSerializeObject("./src/studentTotalMark.txt");

    object.exportToCsv(deSerializeObject);

}
```

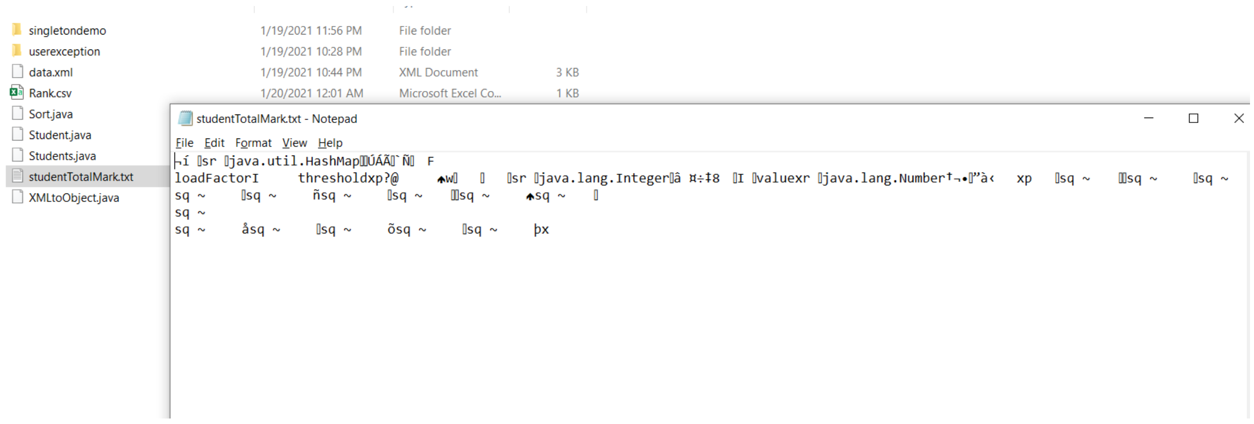
Output:



```java
import javax.xml.bind.*;
public class XMLtoObject {

    //Convert XML file content to Java object using JAXB
    public Map<Integer,Integer> unMarshall() {
        Map<Integer, Integer> studTotal = new HashMap();
        try {
            File file = new File("./src/data.xml");
            JAXBContext jaxbContext = JAXBContext.newInstance(Students.class);
            Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
            Students students = (Students) unmarshaller.unmarshal(file);
            System.out.println("Xml file data converted to java object");
            // Student data in Object
            System.out.println(students);

            List<Student> student = students.getStudent();

            for(Student s:student) {
                studTotal.put(s.getRoll(), studTotal.getOrDefault(s.getRoll(), 0)+s.getMarks());
            }

        } catch (JAXBException e) {
            e.printStackTrace();
        }
        return studTotal;
```

```
<terminated> XMLtoObject [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Jan 20, 2021 12:01:33 AM – 12:01:33 AM)
Xml file data converted to java object
[studentData=[StudentData [roll=11, marks=88, subject=Maths], StudentData [roll=12, marks=85, subject=Maths], StudentData [roll=13, marks=60, subject=Math

The Student Total object was succesfully written to a file

The txt file deserialized to java object
{16=270, 17=269, 18=241, 11=270, 12=266, 13=229, 14=245, 15=254}

Top 5 ranks are exported in CSV file
```

Serialized TXT file

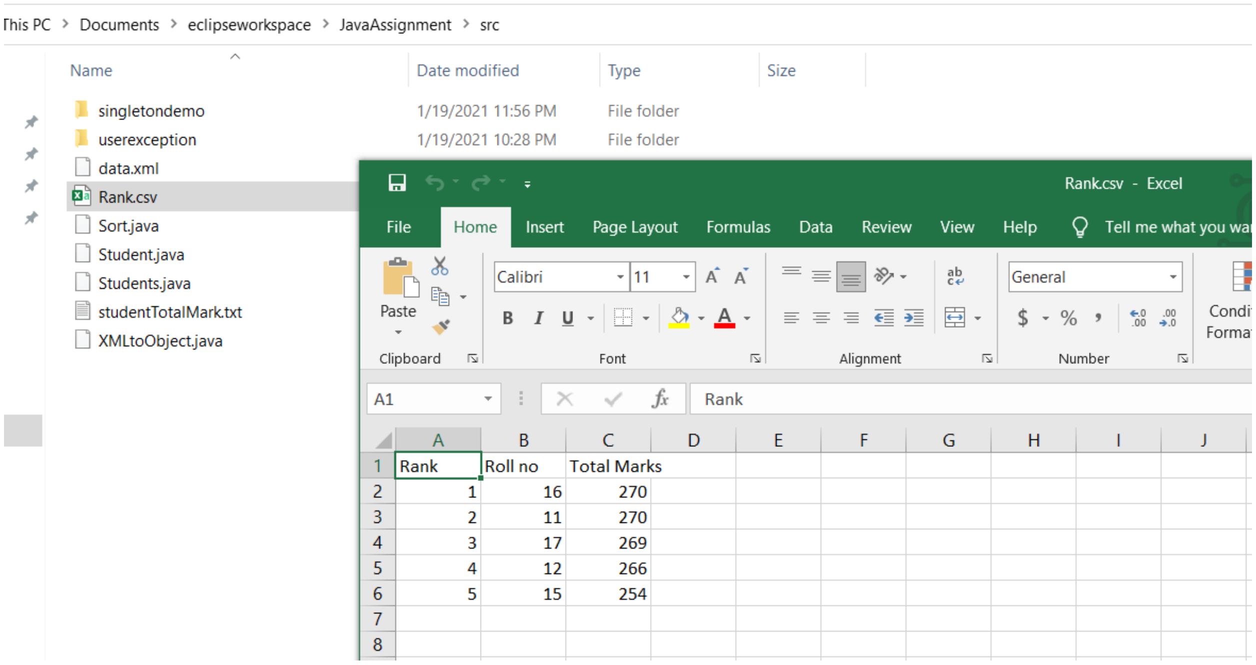


Name	Date modified	Type	Size
singletondemo	1/19/2021 11:56 PM	File folder	
userexception	1/19/2021 10:28 PM	File folder	
data.xml	1/19/2021 10:44 PM	XML Document	3 KB
Rank.csv	1/20/2021 12:01 AM	Microsoft Excel Co...	1 KB
Sort.java			
Student.java			
Students.java			
studentTotalMark.txt			
XMLtoObject.java			

studentTotalMark.txt - Notepad

```
┐í sr java.util.HashMapÚÁÃ`Ñ  F
loadFactorI     thresholdxp?@     wl    sr java.lang.Integerâ ¤÷‡8  I valuexr java.lang.Number†¬•”à‹   xp   sq ~     sq ~     sq ~
sq ~     sq ~     ñsq ~     sq ~     sq ~     sq ~    
sq ~
sq ~     åsq ~     sq ~     õsq ~     sq ~     þx
```

CSV file with top 5 rank



This PC › Documents › eclipseworkspace › JavaAssignment › src

Name	Date modified	Type	Size
singletondemo	1/19/2021 11:56 PM	File folder	
userexception	1/19/2021 10:28 PM	File folder	
data.xml			
Rank.csv			
Sort.java			
Student.java			
Students.java			
studentTotalMark.txt			
XMLtoObject.java			

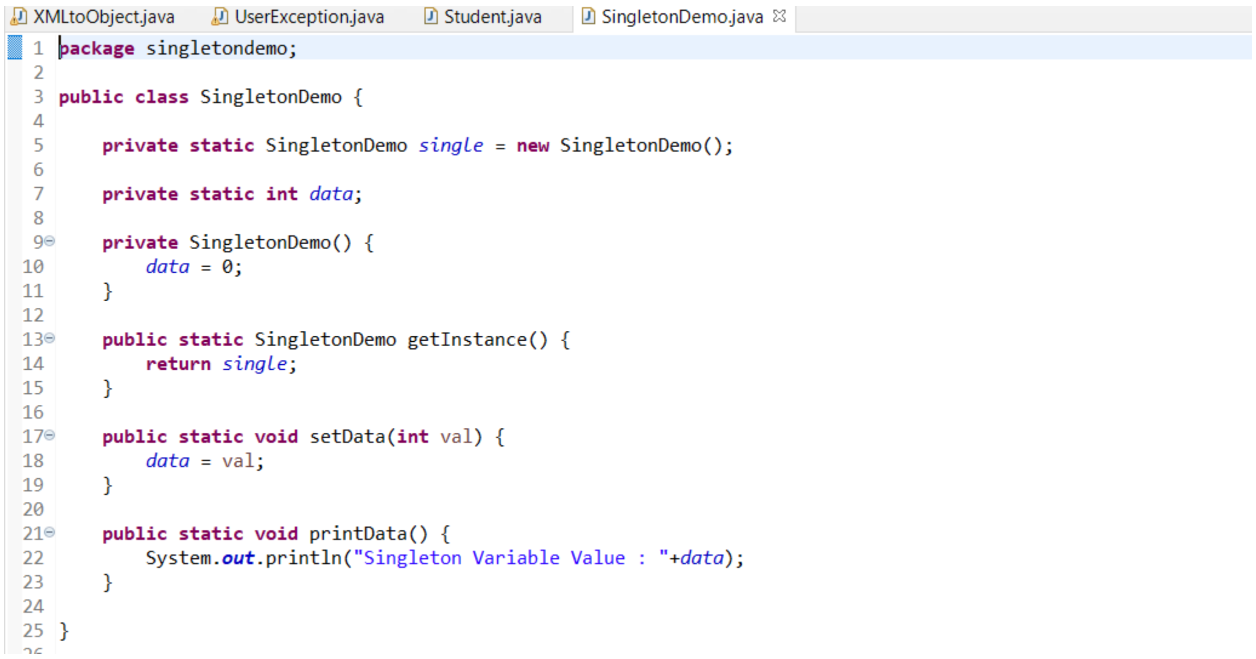
Rank.csv - Excel

A1 Rank

Rank	Roll no	Total Marks
1	16	270
2	11	270
3	17	269
4	12	266
5	15	254

<u>Implementation of singleton class:</u>

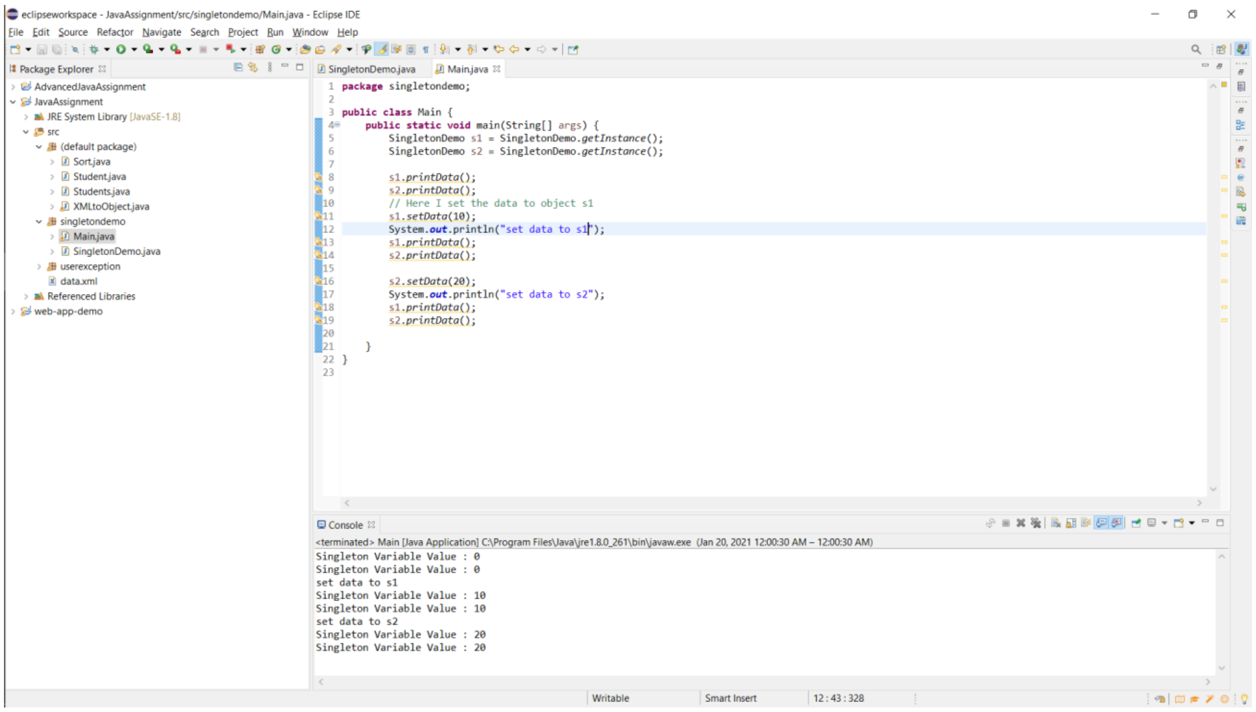
- Basic purpose is control object creation
- Limit the no of objects to only one
- Full of static fields



```java
package singletondemo;

public class SingletonDemo {

    private static SingletonDemo single = new SingletonDemo();

    private static int data;

    private SingletonDemo() {
        data = 0;
    }

    public static SingletonDemo getInstance() {
        return single;
    }

    public static void setData(int val) {
        data = val;
    }

    public static void printData() {
        System.out.println("Singleton Variable Value : "+data);
    }

}
```

Main method and output: you will able to create more than one instance but holds the same reference.

If you change the data in one instance it will reflect in all instance because all fields are static here.



```java
package singletondemo;

public class Main {
    public static void main(String[] args) {
        SingletonDemo s1 = SingletonDemo.getInstance();
        SingletonDemo s2 = SingletonDemo.getInstance();

        s1.printData();
        s2.printData();
        // Here I set the data to object s1
        s1.setData(10);
        System.out.println("set data to s1");
        s1.printData();
        s2.printData();

        s2.setData(20);
        System.out.println("set data to s2");
        s1.printData();
        s2.printData();

    }
}
```

```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe  (Jan 20, 2021 12:00:30 AM – 12:00:30 AM)
Singleton Variable Value : 0
Singleton Variable Value : 0
set data to s1
Singleton Variable Value : 10
Singleton Variable Value : 10
set data to s2
Singleton Variable Value : 20
Singleton Variable Value : 20
```