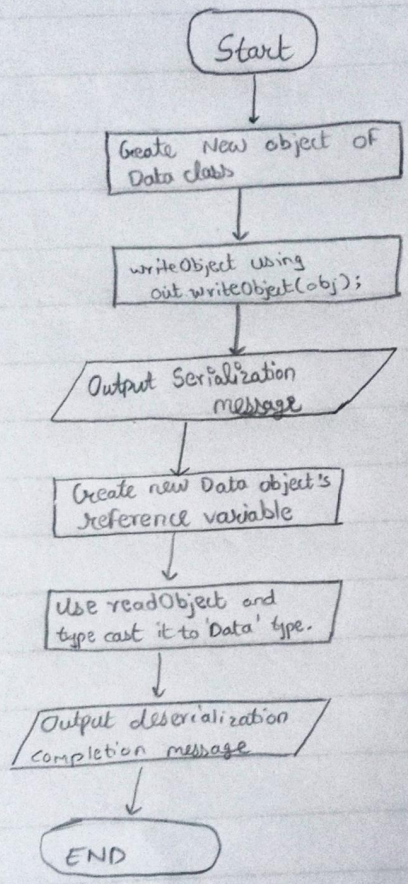
Practical No. 19

Aim : Create, debug and run java programs based on object Serialization.

flowchard :



PAGE NO.
DATE

Practical No. 19

Aim: Create, debug and run java programs based on Object Serialization.

Theory:

What is Object Serialization?

Object serialization is a mechanism provided in Java, using which an object can be represented as a sequence of bytes that includes the object's data as well as information about the object's type and the type's of data stored in the object.

After a serialized object has been written to a file, it can be read from a file and be deserialized, i.e. the type of information and bytes used to store the object can be used to recreate the object in memory.

Most impressive is that the entire process of serialization is JVM independent, that is, an object can be serialised on one platform and can be deserialized from a completely different platform.

The serialization and deserialization of objects is done through classes 'ObjectInputStream' and 'ObjectOutputStream' which are high level streams which contains the methods.

The method used to Serialize an object:

public final void writeObject(Object x) throws IOException

The method used to deserialise an object

public final Object readObject(Object x) throws IOException, ClassNotFoundException.

Conclusion :

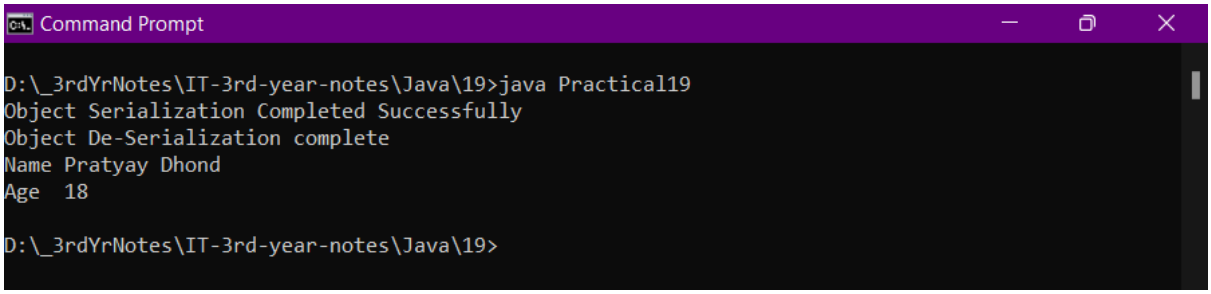
Hence, I create, debugged and executed java programs based on object serialization.

Code:

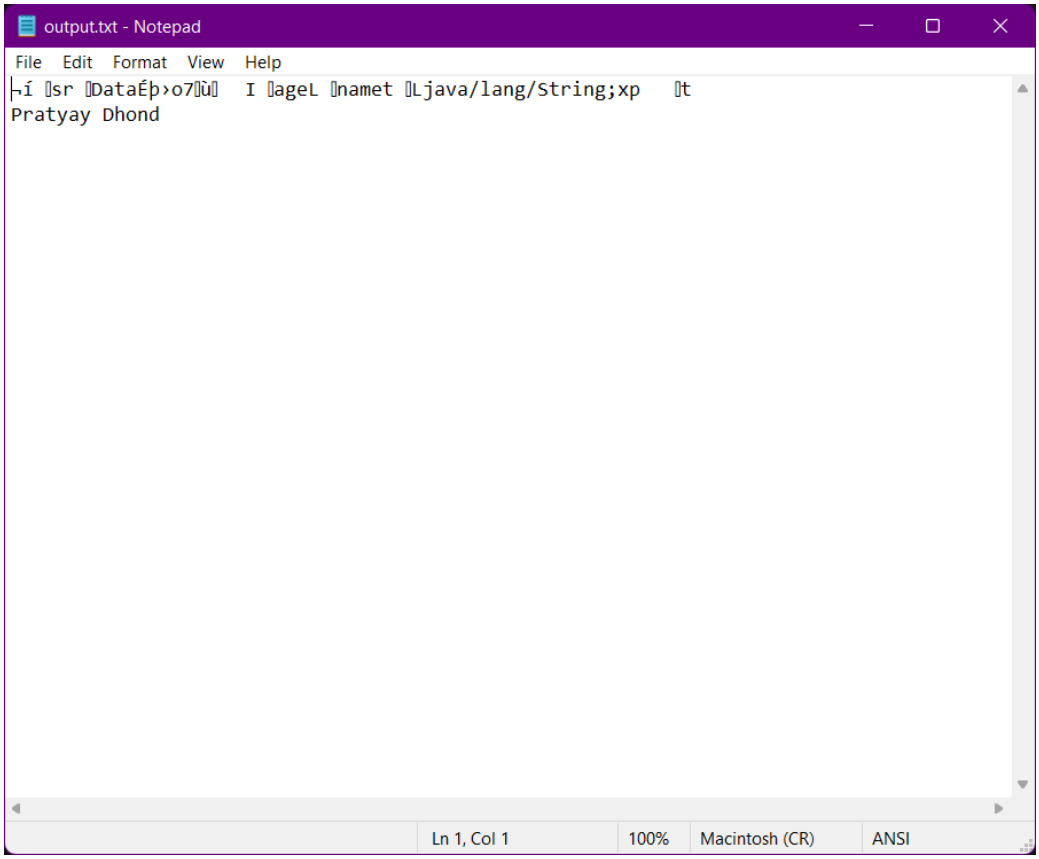
```
import java.io.*;
import java.io.Serializable;

class Data implements java.io.Serializable{
  public String name;
  public int age;

  public Data(String name, int age){
    this.name = name;
    this.age = age;
  }
}

class Practical19{
  public static void main(String[] args){

    Data obj = new Data("Pratyay Dhond",18);
    String filename = "output.txt";

    try{
      FileOutputStream file = new FileOutputStream(filename);
      ObjectOutputStream out = new ObjectOutputStream(file);

      out.writeObject(obj);
      out.close();
      file.close();

    System.out.println("Object Serialization Completed Successfully");

    }catch(Exception e){
          System.out.println("Error : Object Serialization Failed");
          System.out.println("Error : " + e);
    }

    try{
      FileInputStream file = new FileInputStream(filename);
      ObjectInputStream in = new ObjectInputStream(file);

      Data from_file = null;

      from_file = (Data) in.readObject();
      in.close();
      file.close();


      System.out.println("Object De-Serialization complete");
      System.out.println("Name " + from_file.name);
```

```
      System.out.println("Age  " + from_file.age);
    }catch(Exception e){
            System.out.println("Error : Object De-Serialization failed...");
                  System.out.println("Error : " + e);

    }


  }
}
```

Output:



Output on terminal



Serialised data stored in file

PAGE NO.
DATE

Conclusion:

Hence, I created, debugged, and executed java programs based on object serialization.