

```

package SerialTest;

import java.io.*;
// import java.awt.*;

public class MainClass {
    public static void main(String[] args)
    throws IOException, ClassNotFoundException {
        MainClass mainClass = new MainClass();
        mainClass.writeToDisc();
        mainClass.readFromDisc();
    }

    public void writeToDisc() throws IOException, ClassNotFoundException {
        C aC = new C();
        aC.setInt(555);
        A anA = new A();
        anA.setDouble(7.88);
        anA.setInt(888); // sets int of subobject of type B
        anA.setC(aC);
        System.out.println(anA.getDouble());
        System.out.println(anA.getInt());
        System.out.println(anA.getB().getInt());

        double someDoubles[] = new double[5];
        for(int i = 0; i < 5; ++i) {
            someDoubles[i] = i*6.7;
            System.out.print(someDoubles[i]+":");
        }
        System.out.print("\n");

        // Write Object anA to file myfile.ser. Subobject of type B
        // is automatically saved with its state
        FileOutputStream aFOS = new FileOutputStream("myfile.ser");
        ObjectOutputStream anOOS = new ObjectOutputStream(aFOS);
        anOOS.writeObject("Here is the preserved data:");
        anOOS.writeObject(anA);
        anOOS.writeObject(someDoubles);
        anOOS.flush();
        aFOS.close();
    }

    public void readFromDisc()throws IOException, ClassNotFoundException {
        // Read and create object of type A. Subobject of type B is automatically
        // read and created
        FileInputStream aFIS = new FileInputStream("myfile.ser");
        ObjectInputStream anOIS = new ObjectInputStream(aFIS);
        String aString = (String)anOIS.readObject();
        A anotherA = (A)anOIS.readObject();
        double[] anotherSomeDoubles = (double[])anOIS.readObject();
        aFIS.close();

        System.out.println(aString);
        System.out.println(anotherA.getDouble());
        System.out.println(anotherA.getInt());
        B anotherB = anotherA.getB();
        System.out.println(anotherB.getInt());
        C anotherC = anotherA.getC();
        System.out.println(anotherC.getInt());
        for(int i = 0; i < 5; ++i) {
            System.out.print(anotherSomeDoubles[i]+":");
        }
    }
}

```

```
    }  
}  
}  
  
class A implements Serializable {  
    private B itsB = null;  
    private C itsC = null;  
    private double itsDouble = 0.0;  
  
    public A() { itsB = new B(); }  
    public double getDouble(){ return itsDouble; }  
    public void setDouble(double aDouble) { itsDouble = aDouble;}  
    public void setInt(int anInt){ itsB.setInt(anInt); }  
    public int getInt() { return itsB.getInt();}  
    public B getB() { return itsB; }  
    public void setC(C aC) { itsC = aC; }  
    public C getC() {return itsC; }  
  
}  
  
class B implements Serializable{  
    private int itsInt = 0;  
    public int getInt(){ return itsInt; }  
    public void setInt(int anInt) { itsInt = anInt;}  
  
}  
  
class C implements Serializable{  
    private int itsInt = 0;  
    public int getInt(){ return itsInt; }  
    public void setInt(int anInt) { itsInt = anInt;}  
}
```

```
/* --- output ---  
7.88  
888  
888  
0.0:  
6.7:  
13.4:20.1:26.8:  
Here is the preserved data:  
7.88  
888  
888  
555  
0.0:6.7:13.4:20.1:26.8:  
  
--- end output ---  
*/
```