

Module 6—Working with Files

1. Write a programme to search non-recursively for filenames where the user will give the following inputs as command line arguments

- The search directory
- The file extension

2. Write a method in the module6.TextFileHandler to load the file into a map, where key = country and value = capital.

3. Write a method in the module6.TextFileHandler to write, to a file, all the country names that starts with a letter passed as an argument, along with their capitals.

•**public static void writeToFile(String newFilename, char countryNameBeginningWith)** where **countryNameBeginningWith = 'S'**

•**Name the new file countries_s.csv**

4. For the class module6.SerializationDemo

•Comment out the **load(..)** method call and just store the list by running the programme.

•Now change the **serialVersionUID** to something else.

•Comment the **store(..)** method call, uncomment the **load(...)** method call and run the programme. What do you observe?

SerializationDemo.java

```
package module6;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.List;
import java.util.Vector;

/**
 * Demonstrates serialization and deserialization
 * @author pchandra
 */
public final class SerializationDemo
{
    /**
     * Persists a list of {@link Person} to a file.
     * @param list
     * @param filename
     * @throws FileNotFoundException
     */
}
```

```

    * @throws IOException
    */
    public static void store(List<Person> list, String filename) throws
FileNotFoundException, IOException
    {
        try(ObjectOutputStream outstream =
            new ObjectOutputStream(new FileOutputStream(filename));)
        {
            outstream.writeObject(list);
        }
    }

    /**
     * Loads a list of {@link Person} from a file
     * @param filename
     * @return
     * @throws FileNotFoundException
     * @throws IOException
     * @throws ClassNotFoundException
     */
    @SuppressWarnings("unchecked")
    public static List<Person> load(String filename)
        throws FileNotFoundException, IOException, ClassNotFoundException
    {
        try(ObjectInputStream instream =
            new ObjectInputStream(new FileInputStream(filename));)
        {
            return (List<Person>)instream.readObject();
        }
    }

    public static void main(String[] args)
    {
        if (args.length != 1)
        {
            System.out.println("Usage: java module6.SerializationDemo
filename");
            System.exit(-1);
        }

        // Getting the list ready
        Person p1 = new Person("Karl Pearson", new Integer(79), "United
Kingdom"),
            p2 = new Person("John Tukey", new Integer(85), "United
States");

        Vector<Person> inList = new Vector<Person>();
        inList.add(p1);
        inList.add(p2);

        try
        {
            // Store the list to a file
            SerializationDemo.store(inList, args[0]);
        }
    }

```

```

        // Load the list from a file
        List<Person> outList = SerializationDemo.Load(args[0]);
        for (Person p : outList)
            System.out.println(p.toString());
    }
    catch (IOException | ClassNotFoundException e)
    {
        e.printStackTrace();
    }
}

class Person implements Serializable
{
    /**
     * Generated from Eclipse IDE
     */
    private static final long serialVersionUID = -5551642649093889857L;

    /**
     * Persitable fields
     */
    protected String name = null;
    protected Integer age = null;
    /**
     * Non-persitable fields
     */
    protected transient String location = null;

    public Person() {}

    public Person(String name, Integer age, String location)
    {
        this.name = name;
        this.age = age;
        this.location = location;
    }

    @Override
    public String toString()
    { return "Name: " + name + " Age: " + age + " Location: " + location; }
}

```