The Tutorial Questions and Lab Projects of Week 2

Tutorial Questions

1. A server creates a port which it uses to receive requests from clients. Discuss the design issues concerning the relationship between the name of this port and the names used by clients.

2. Sun XDR marshals data by converting it into a standard big-endian form before transmission. Discuss the advantages and disadvantages of this method when compared with CORBA's CDR.

3. Why is there no explicit data-typing in CORBA CDR?

4. Write an algorithm in pseudocode to describe the serialization procedure. The algorithm should show when handles are defined or substituted for classes and instances. Describe the serialized form that your algorithm would produce when serializing an instance of the following class Couple.

```
class Couple implements Serializable{
       private Person one;
       private Person two;
       public Couple(Person a, Person b) {
       one = a;
       two = b;
       }
}
```

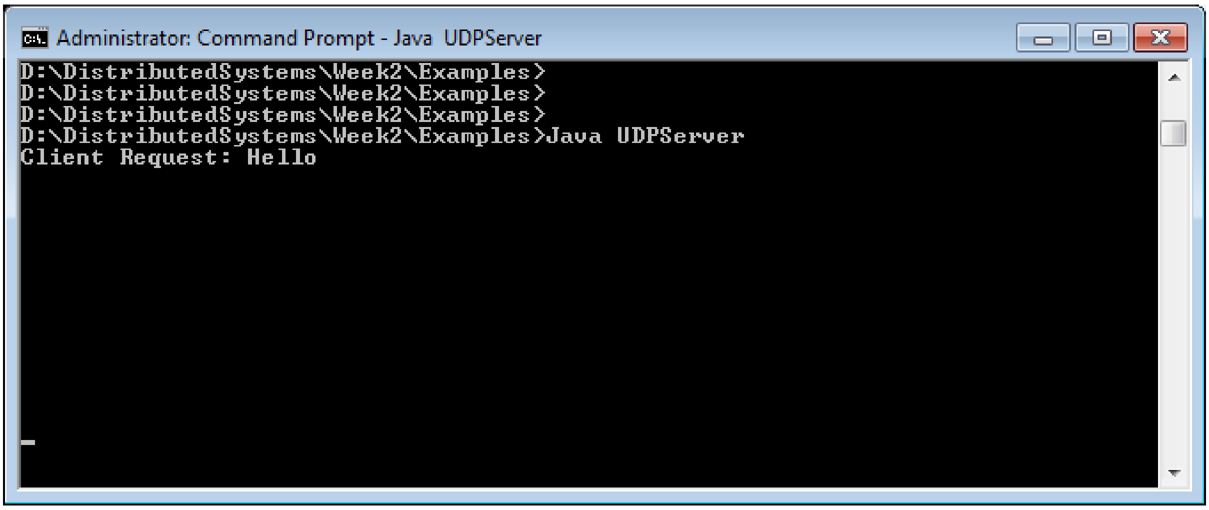
5. Outline how connectionless communication between a client and a server is established and proceeded by using sockets.

Lab Projects

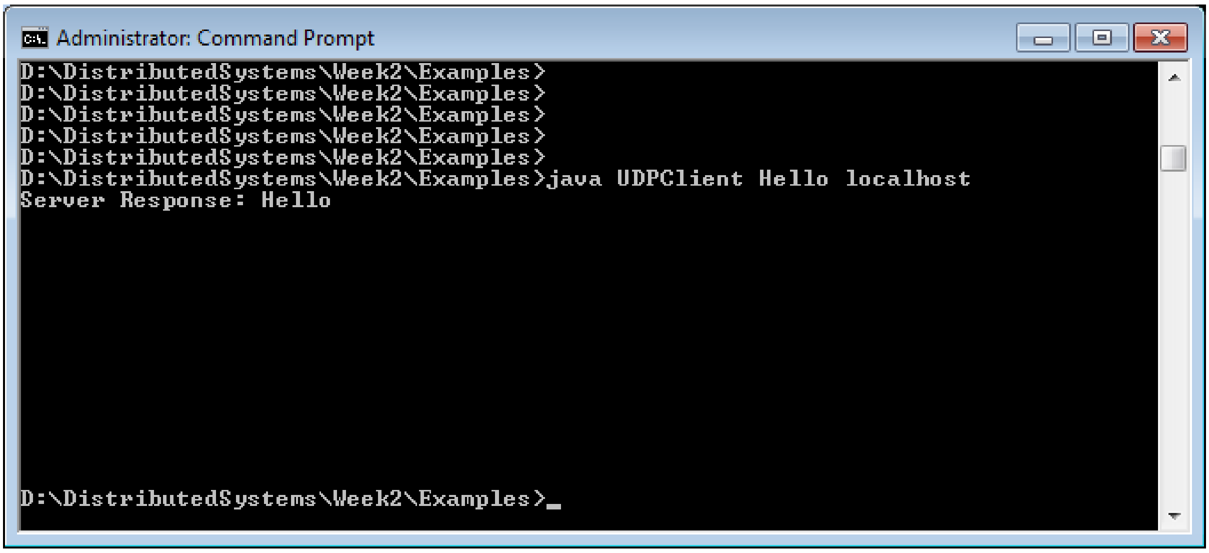
On the lecture slides of Week 2, there is client/server example using UDP protocol. This example consists of programs: `UDPClient.java` and `UDPserver.java`.

Task 1:

1. Revise the two programs so that the outputs of the programs like the following screenshots.
2. Compile the two programs to generate Java class files: `UDPClient.class` and `UPDServer.class`.
3. Run the two java classes on two JVMs (Java Virtual Machines) on a single physical machine and check the program outputs.



```
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>Java UDPServer
Client Request: Hello
```

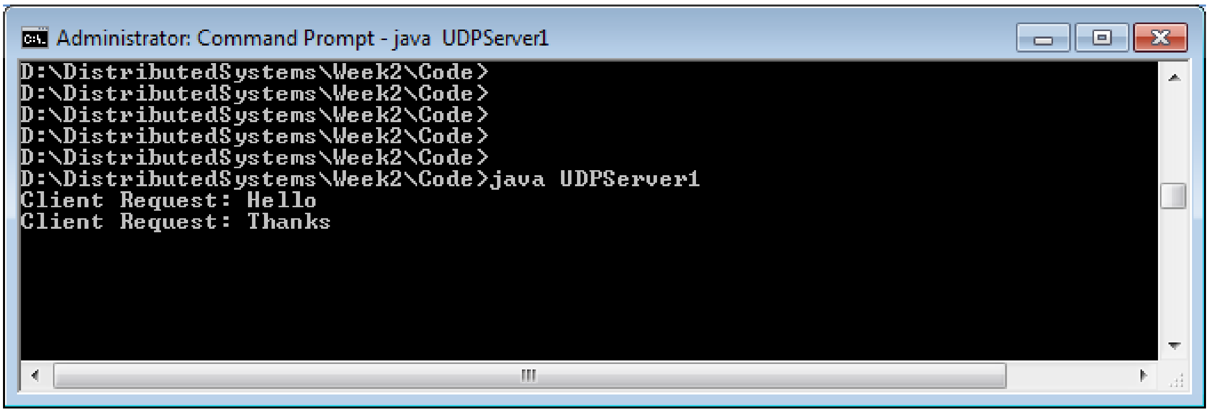


```
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>
D:\DistributedSystems\Week2\Examples>java UDPClient Hello localhost
Server Response: Hello












D:\DistributedSystems\Week2\Examples>_
```

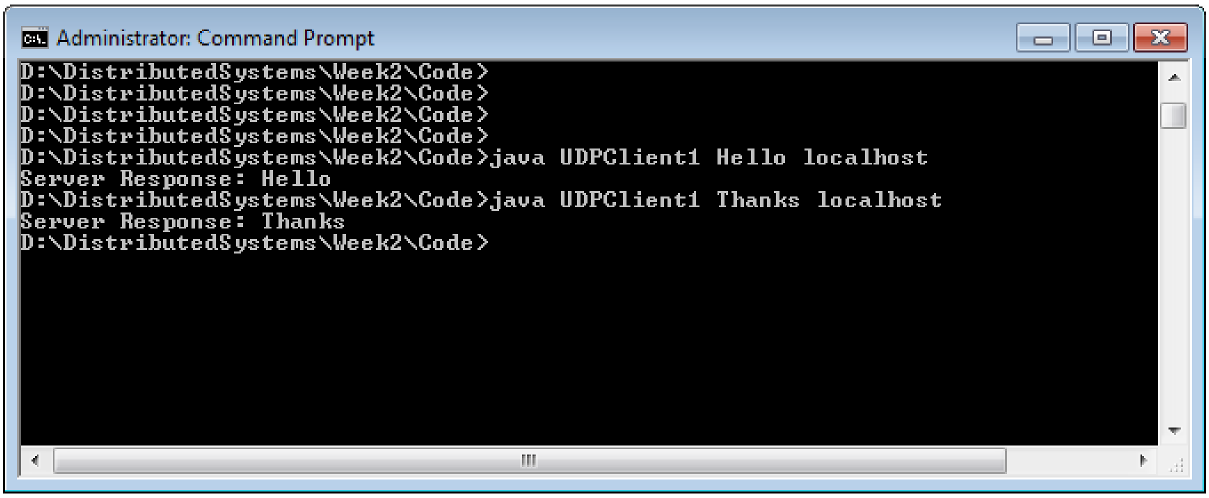
Task2:

From the outputs of `UDPClient` or `UDPServer`, you have seen that there are many blank lines for each request or response.

1. Investigate why there are blank lines in the outputs.
2. Revise the two programs: `UDPCLient.java` and `UDPserver.java` so that the programs just output the strings of the request or the response but remove the blank lines. The outputs of the revised programs should like those from `UDPCLient1` and `UDPserver1` as in the following screenshots.
3. Compile the revised programs, run them and check the outputs.



```
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>java UDPServer1
Client Request: Hello
Client Request: Thanks
```



```
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>java UDPClient1 Hello localhost
Server Response: Hello
D:\DistributedSystems\Week2\Code>java UDPClient1 Thanks localhost
Server Response: Thanks
D:\DistributedSystems\Week2\Code>
```

Note:

To make the above revision, you need to investigate Java APIs at least for the class: `String` and `DatagramPacket`. The reference is the Java APIs document from Sun Microsystems. It is online at:

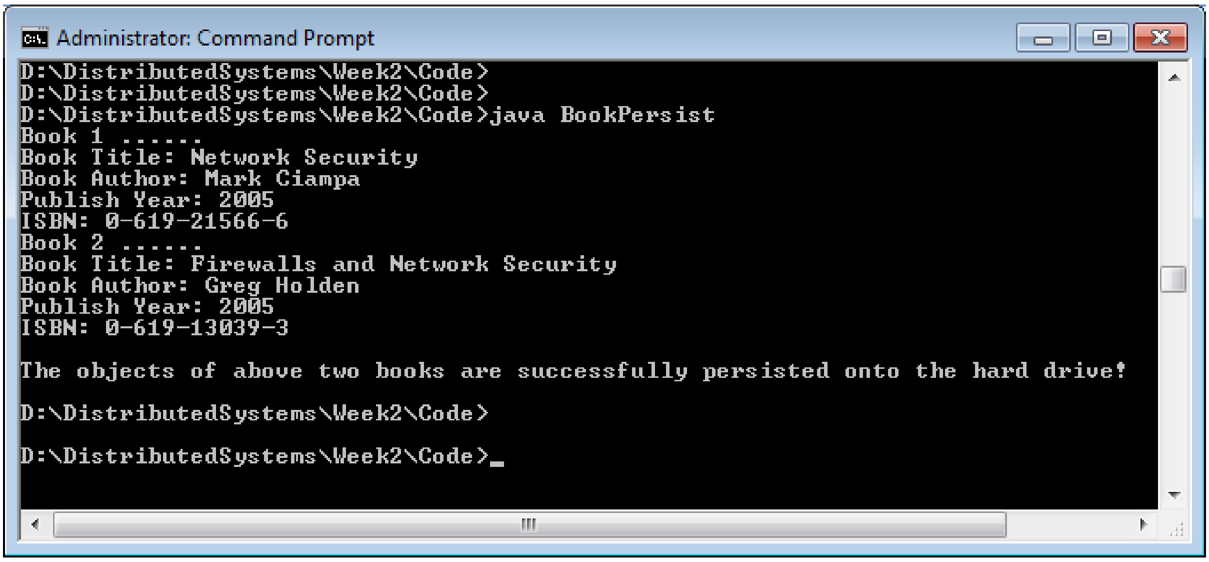
http://docs.oracle.com/javase/6/docs/api/

Task 3:

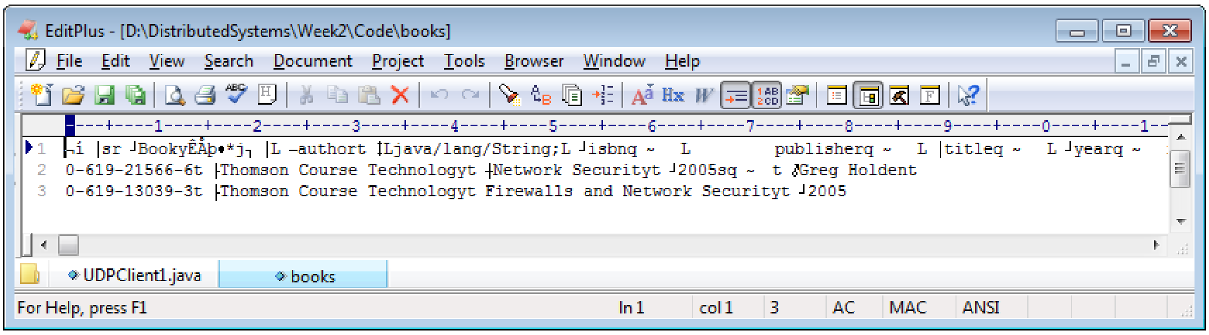
Define a **Book** class, which has the fields: `Title`, `Author`, `Publisher`, `PublishYear` and `ISBN`. The class should have methods for getting value from and setting value onto each field.

Write a program to create two objects from the Book class and save the objects into a file onto permanent storage by using Java serialization.

The following are the screenshots of the output of such a program and the file that stores the Book objects.



```
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>
D:\DistributedSystems\Week2\Code>java BookPersist
Book 1 .......
Book Title: Network Security
Book Author: Mark Ciampa
Publish Year: 2005
ISBN: 0-619-21566-6
Book 2 .......
Book Title: Firewalls and Network Security
Book Author: Greg Holden
Publish Year: 2005
ISBN: 0-619-13039-3

The objects of above two books are successfully persisted onto the hard drive!

D:\DistributedSystems\Week2\Code>

D:\DistributedSystems\Week2\Code>_
```



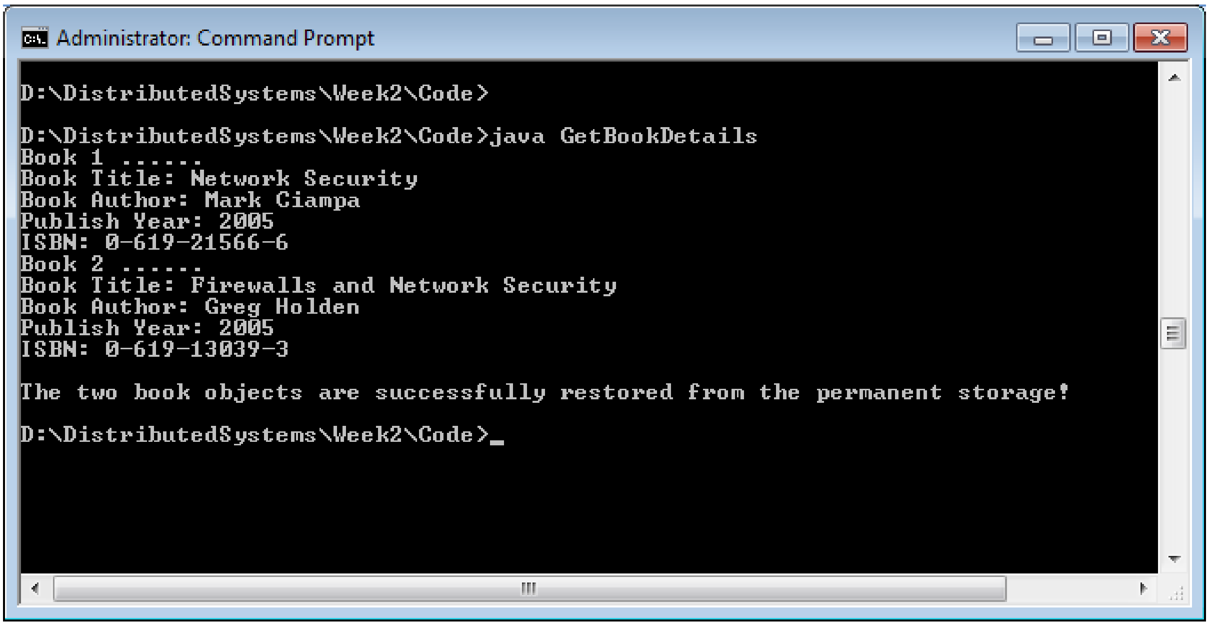
EditPlus - [D:\DistributedSystems\Week2\Code\books]
File Edit View Search Document Project Tools Browser Window Help
UDPClient1.java
books
For Help, press F1
ln 1
col 1
3
AC
MAC
ANSI

```
┐i |sr ┘BookyÊÅþ•*j┐ |L –authort ‡Ljava/lang/String;L ┘isbnq ~  L        publisherq ~  L |titleq ~  L ┘yearq ~
0-619-21566-6t |Thomson Course Technologyt ┤Network Securityt ┘2005sq ~  t ♂Greg Holdent
0-619-13039-3t |Thomson Course Technologyt Firewalls and Network Securityt ┘2005
```

Task 4:

Write another program to read the file from the hard drive and restore the two objects of Book. After that, print the publishing information for each book to confirm the serialization and deserialization are successful.

The following is the screenshot of the outputs of such a program.



```
D:\DistributedSystems\Week2\Code>

D:\DistributedSystems\Week2\Code>java GetBookDetails
Book 1 ......
Book Title: Network Security
Book Author: Mark Ciampa
Publish Year: 2005
ISBN: 0-619-21566-6
Book 2 ......
Book Title: Firewalls and Network Security
Book Author: Greg Holden
Publish Year: 2005
ISBN: 0-619-13039-3

The two book objects are successfully restored from the permanent storage!

D:\DistributedSystems\Week2\Code>_
```

The reference resources for this project are:

1. Week 2 lecture slides

2. Java object serialization specification from Sun Microsystems at:

   http://docs.oracle.com/javase/1.3/docs/guide/serialization/

3. Java IO APIs on `FileOutputStream`, `ObjectOutputStream`, `FileInputStream`, and `ObjectInputStream`.

   http://docs.oracle.com/javase/6/docs/api/