1. 說明OSI網路7層架構，各層的核心概念與資料單位是什麼? (10分)

**Physical Layer:**電線或物理材料上傳遞0或1電子訊號，EX. 電導體、光纖、空氣……訊號透過編碼將物理訊號轉換為電腦可以接受的資料 (0 與 1) EX. (電) 5V => 1; 0V => 1 ，或是透過轉換將不同的通訊材料間訊號的轉換光<=>電。

**Data Link Layer**:確保點對點的聯接的資料的正確性，網路之間建立邏輯連結，傳輸過程中流量控制以及錯誤偵測。設備包括switch=>將網路Data依Mac位址傳至目的位址。

**Network Layer**:網路路由以及定址功能，最佳路徑的選擇。IP 屬於這一層，包過來源以及目的，加入傳送Data=>Packet。設備包括路由器。

**Transport Layer**: 端對端的資料傳送，電腦整體的資料傳輸以及控制(5/6/7layer流量控制以及錯誤偵測)=>大量資料切割

二個重要的協定:

* TCP: Transmission Control Protocol
  + 可靠的通訊模式的 (資料遺失，會重送)
  + 連接導向式通訊 (connection-oriented Communication)
  + 一次可以送多個封包 (packet，資料傳送的單位)
  + 可以做流量控制 (透過滑動窗，Sliding Window，決定傳送多封包)
  + 有錯誤偵測&糾正方法
  + 收到後會將確認送回發送者
  + 慢
  + EX. web, mail, ssh
* UDP: User Datagram Protocol
  + 非可靠的通訊模式 (資料遺失，不會重送)
  + 非連接導向式通訊 (connectionless Communication)
  + 一次只能送一個封包
  + 不能做流量控制
  + 沒有錯誤偵測&糾正方法
  + 不會將收到後會將確認送回發送者
  + 快
  + EX. 音頻, DNS, 在線遊戲

**Session Layer**:網路連線 => Data transport => Disconnent，處理通訊方式的正確選擇

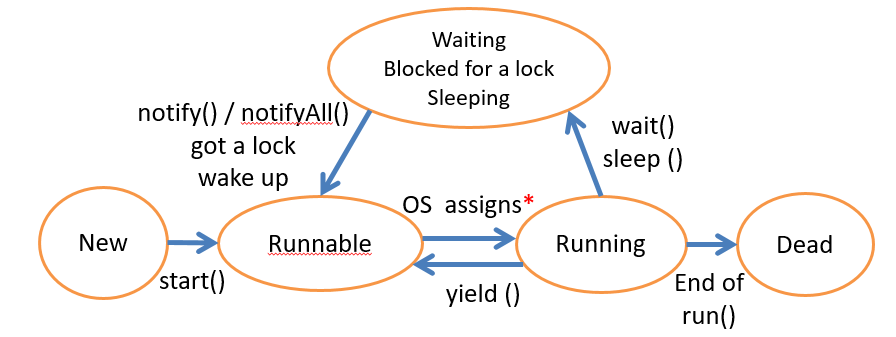
* + 單工：只能單方向傳或收資料，例如電視
  + 半雙工：可傳或收資料，但同一時間只能在傳與送資料之中，選擇一種，例如：有些對講機
  + 全雙工:可同時傳或收資料，例如：講電話

**Presentation Layer**: 語意的正確解釋，在網路通訊中，這一層不一定需要，如果沒加解密、沒資料壓縮/解壓縮，沒編碼/解碼正確性的問題

* + 資料加密/解密
  + 資料壓縮/解壓縮
  + 編碼/解碼的正確性

**Application Layer**: 應用軟體或是協定，提供某種網路的服務，網路下載功能

1. 請用狀態圖說明 Thread 的機制？(10分)



可透過繼承 Thread

* + class T1 extends Thread
  + override run()

或實作 Runnable，並導入到 Tread 物件

* + class MyThread implements Runnable { …實作 run() }
  + Thread T2 = new Thread (MyThread)

wait() : 將 該物件的 lock 交出去，本身進入 waiting 狀態

notify():通知一個等待某物件 lock 的 thread，讓該 thread 進入runnable 狀態， 但不能保證那個 thread，可以進入 running 狀態。

notifyAll(): 通知全部等待某物件的 thread 進入runnable 狀態，但只會有一個thread 進入running 狀態， 不能指定哪個特定的 thread 進入running 狀態。

join(): 等到特定的 tread 失效後，再接續處理。

sleep(): static 方法，讓一個thread 睡一特定時間，睡醒後該thread 會進入runable 狀態

yield(): static 方法，將現在running 的 thread 變成在 runnable 狀態。

其中可透過.run() 或 .start() , run()可重複呼叫 .start()則否。

1. 何謂 Thread 的 synchronized?它的方法有哪些? (10分)

synchronized block

synchronized(object){ … }

synchronized method

synchronized public void m1(){ … }

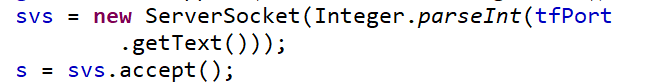
等同於：

public void m1(){

synchronized (this){ … }

}

1. 說明 TCP socket 建立與使用的流程，並以程式碼說明之。(10分)
2. Server 端建立 ServerSocket，註明 IP/URL + Port number。
3. Server 端用ServerSocket.accept()來取得和Client端連線的Socket物件。

ex sever端先建立監聽

若沒接收到Client端得連線請求則不會到accept

1. Client 端產生 socket 物件，並自動聯上 Server。

Ex.

1. 當Server和Client連接後，Socket 就可以用getInputStream() 取得 InputStream 物件跟用getOutputStream() 取得 OutputStream 物件，執行 Server/Client 間的資料傳送與接收，必要時可以用 InputStreamReader、OutputStreamWriter、BufferedReader 、BufferedWriter 等物件 。
2. 資料傳遞完後，記得要呼叫Socket 的close()，來關閉所有使用到的Sockets。

Ex. s.close();

1. 說明如何利用socket傳送 Object，並以程式碼說明之。(10分)

透過OutputStream以及InputStream來傳送以及接收資料

OutputStream out = gui.s.getOutputStream();

InputStream in = gui.s.getInputStream();

1. 說明 SMTP、IMAP、POP3。 (20分)

**SMTP**: 一種在 internet 傳送 email 的標準，在其之上指定了一條訊息的一個或多個接收者（在大多數情況下被確認是存在的），然後訊息文字會被傳輸，使電子郵件能自由的在各個郵件伺服器上傳送。可以很簡單地通過telnet程式來測試一個SMTP伺服器。使用TCP埠25。一開始只支援 ASCII，但現在可以支援8位元MIME擴充功能。郵件外送需指定 SMTP server，例如：smtp.nknu.edu.tw，沒有對傳送方做身分驗證，如要做身分驗證，可使用 SMTP-AUTH。SMTP收信時並無法自由的將信件從伺服器上擷取下來，使用者必須自己再到郵件伺服器上收取信件，所以後來的郵件軟體便增加了POP3的功能，相對的，郵件伺服器上也多了POP3的協定。

**POP3**: POP處理過程：郵件傳送到伺服器上，電子郵件用戶端呼叫郵件用戶端程式以連線伺服器，並下載所有未閱讀的電子郵件。一旦郵件傳送到用戶端電腦，郵件伺服器上的郵件將會被刪除。用於接收 email (取信)，常用於用戶端電腦，例如：用戶端的Outlook上可使用POP3，接收 email。適合只在一台機器上使用 email。Off-line 使用，須定時連上伺服器，下載信件。POP3主要是作為收信時的協定，有了POP3的功能，使用者只要一開郵件軟體，伺服器上的信件便會自動的被擷取到自己的電腦上，而不須再多費功夫到主機上去尋找，所以到後來收發E-mail的協定便分別由SMTP（發送信件）和POP3（收取信件）分別負責。

IMAP: IMAP和POP3不同的地方是在於收取信件的方式；POP3的方式是將伺服器上所有的信件一併擷取到自己的電腦上，缺點就是會不分郵件性質、是否要接收的全部抓下來，匯入使用的電腦中。因此當使用者在外地，卻需收取E-mail時，便無從選擇的會將所有信件抓到別台電腦；而IMAP就解決了這個缺點，它可直接在主機上編輯郵件，也就是直接在伺服器上看信件內容，再決定是否要將信件抓下來，有Remote（線上編輯）和Local（離線編輯）兩種模式，Local和POP3類似，只是Local可自由選擇所要下載的信件。也可讓使用者以Web Mail 方式，讀取信件。適合需要在多台機器上使用 email。On-line 使用，可即時收信。

7. 何謂MIME (Multipurpose Internet Mail Extensions)，它有什麼功用? 請列舉出三個 MIME 的 sub-types。 (10分)

MIME並不是用於傳送郵件的協議，它作為多用途郵件的擴充套件定義了郵件內容的格式：資訊格式、附件格式等等。是一個網際網路標準，它擴展了電子郵件標準，使其能夠支援：

* + 非ASCII字符文本
  + 非文本格式附件（二進位、聲音、圖像等）
  + 由多部分（multiple parts）組成的消息體
  + 包含非ASCII字符的頭信息（Header information）。

內容類型（Content-Type），這個頭部領域用於指定消息的類型。一般以下面的形式出現。Content-Type: [type]/[subtype]; parameter。Ex. Text、Message、Audio……

subtype用於指定type的詳細形式。content-type/subtype配對的集合和與此相關的參數，將隨著時間而增長。為了確保這些值在一個有序而且公開的狀態下開發，MIME使用Internet Assigned Numbers Authority (IANA)作為中心的註冊機制來管理這些值。Ex. text/plain（純文本）、text/html（HTML文檔）、application/xhtml+xml（XHTML文檔）……

parameter可以用來指定附加的信息，更多情況下是用於指定text/plain和text/htm等的文字編碼方式的charset參數。MIME根據type制定了默認的subtype，當客戶端不能確定消息的subtype的情況下，消息被看作默認的subtype進行處理。

8. 說明如何利用java mail API，實作寄信與收信軟體，並以程式碼說明之。(20分)

javax.mail的包下面存在著一些核心類：Session、Message、Address、 Authenticator、Transport、Store、Folder。

**Session**類定義了基本的郵件會話。就像Http會話那樣，進行收發郵件的工作都是基於這個會話的。Session物件利用了java.util.Properties物件獲得了郵件伺服器、使用者名稱、密碼資訊和整個應用程式都要使用到的共享資訊。

Properties props = new Properties();

Session session = Session.getDefaultInstance(props, null);

Or

Session session = Session.getInstance(props, null);

**Message**: 建立了Session物件後，便可以被髮送的構造資訊體了。在這裡SUN提供了Message型別來幫助開發者完成這項工作。由於Message是一個抽象類，大多數情況下，使用javax.mail.internet.MimeMessage這個子類，該類是使用MIME型別、MIME資訊頭的郵箱資訊。

message.setContent("Hello", "text/plain");

**Address**: ，Address類也是一個抽象類，所以我們將使用javax.mail.internet.InternetAddress這個子類。

Address address = new InternetAddress("test@gmail.com ");

通過message的setFrom()和setReplyTo()兩種方法設定郵件的發信人

message.setFrom(address);message.setReplyTo(address)

or

Address address[] = ...;message.addFrom(address);

Authenticator: JavaMail API通過使用授權者類以使用者名稱、密碼的方式訪問那些受到保護的資源，在這裡“資源”就是指郵件伺服器。在javax.mail包中可以找到這個JavaMail的授權者類。

Properties props = new Properties();

Authenticator auth = new MyAuthenticator();

Session session = Session.getDefaultInstance(props, auth);

Transport: 這個類實現了傳送資訊的協議（通稱為SMTP），此類是一個抽象類，用send()來傳送訊息

Transport.send(message);

Store和Folder: 從Session中獲取特定型別的Store，然後連線到 Store，這裡的Store代表了儲存郵件的郵件伺服器。在連線Store的過程中，極有可能需要用到使用者名稱、密碼或者Authenticator。

Store store = session.getStore("pop3");

store.connect(host, username, password);

連線到Store後，一個Folder物件即目錄物件將通過Store的getFolder()方法被返回，可從這個Folder中讀取郵件資訊:

Folder folder = store.getFolder("INBOX");

folder.open(Folder.READ\_ONLY);

Message message[] = folder.getMessages();

在讀取郵件時，可以用Message類的getContent()方法接收郵件或是writeTo()方法將郵件儲存，getContent()方法只接收郵件內容（不包含郵件頭），而writeTo()方法將包括郵件頭。

System.out.println(((MimeMessage)message).getContent());

最後關閉Folder和Store

folder.close(aBoolean);

store.close();

send ex.

Properties props = System.getProperties();

props.put("mail.smtp.host", host);

Session session = Session.getDefaultInstance(props, null);

MimeMessage message = new MimeMessage(session);

message.setFrom(new InternetAddress(from));

message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

message.setSubject("Hello JavaMail");

message.setText("Welcome to JavaMail");

Transport.send(message);

Reseive ex.

Properties props = new Properties();

Session session = Session.getDefaultInstance(props, null);

Store store = session.getStore("pop3");

store.connect(host, username, password);

Folder folder = store.getFolder("INBOX");

folder.open(Folder.READ\_ONLY);

Message message[] = folder.getMessages();