

所別：資訊管理學系碩士班 組 科目：計算機概論

一、單選題 (1% each)

1. PCM(Pulse Code Modulation)是屬於下列哪一種轉換？  
(1) 數位至類比，(2) 數位至數位，(3) 類比至數位，(4) 類比至類比
2. 下列哪一種通信編碼方式不提供同步功能？  
(1) NRZ-L，(2) RZ，(3) NRZ-I，(4) Manchester
3. ICMP 訊息是封裝(encapsulated)到下列哪一種資料格式中？  
(1) HDLC frame，(2) Ethernet frame，(3) IP datagram，(4) TCP packet
4. 當 Ethernet 網路卡偵測到一個碰撞(collision)時，會採取下列哪一項動作？  
(1)立刻重送 Ethernet frame，(2) 電腦送出一個壅塞(jam)信號，(3) backoff 值重設為 0，(4) backoff 值減一
5. 一般網路惡意程式(malicious program)所使用的緩衝區溢位(buffer overflow)攻擊是指攻擊電腦裡的哪一部份？  
(1) CPU 裡的控制單元，(2) 磁碟中的 DLL 檔案，(3)記憶體中的堆疊區域，(4) 記憶體控制晶片

二、現代的電腦幾乎都需要用到快取記憶體(cache memory)以及虛擬記憶體(virtual memory)，請針對這兩種記憶體扼要回答下列問題：

1. (2%)快取記憶體的用途為何？
2. (6%)快取記憶體會用到 write-through 與 write-back 兩種技術，請比較這兩種技術的優缺點。
3. (2%)虛擬記憶體的用途為何？
4. (4%)請畫圖簡述分頁(paging)虛擬記憶體的工作原理？

三、在資料通訊中，CRC (Cyclic Redundancy Check) 編碼的用途與工作原理為何？請扼要說明(4%)。今已知 data 內容為 1010011110, divisor 為 1011, 請問產生出來的 CRC code 為何(2%)？必須寫出計算式及答案才計分。

四、填充題 (以英文作答為佳，2% each)

1. \_\_\_\_\_ are a series of steps that help a software team to understand and manage uncertainty.
2. In UML, \_\_\_\_\_ and collaboration diagram are interaction diagrams used to describe the behaviors among a group of objects.
3. The Software Engineering Institute (SEI) has developed a comprehensive model predicted on a set of software engineering capabilities that should be present as organizations reach different levels of \_\_\_\_\_.
4. \_\_\_\_\_ occurs when one module makes use of data or control information maintained within the boundary of another module.
5. An \_\_\_\_\_ approach, called evolutionary prototyping, uses the prototype as the first part of an analysis activity that will be continued into design and construction. The prototype of the software is the first evolution of the finished system.

五、(5%) 試對下面的專案活動 (project activities)，畫出一完成時間最短的 CPM diagram，並標示 critical path。(在答案上，以題目裏敘述的代號標示 activities 即可，如以 A1 標示 "X program coding" 之 activity。)

- A1: X program coding，需時 3 天。
- A2: X program unit test，需時 5 天。
- A3: Y program coding，需時 5 天。
- A4: Y program unit test，需時 2 天。
- A5: Z program coding，需時 7 天。
- A6: Z program unit test，需時 3 天。
- B1: Database design，需時 3 天，output 為 X program 開始 coding 所需。
- B2: Beta test，需時 7 天。
- B3: Alpha test，需時 7 天。
- B4: System test，需時 5 天。

注意：背面有試題

- 六、(5%) 在資料倉儲 (data warehouse) , OLAP 產品 (product) 的功能為何？
- 七、(5%) 試以圖式加上必要之指示說明方式，描繪在物件導向資料庫 (object oriented database) 裏，一個物件的成份、特性 (component and characteristics) 及一切相關事務？

- 八、(15%) 在下列排序程式中，若要用此演算法對 n 個資料加以排序，請回答下列問題：

甲、最好的情況下，需要請個 comparison operations ?  
乙、最好的情況下，需要請個 swap operations ?  
丙、平均的情況下，需要請個 comparison operations ?  
丁、平均的情況下，需要請個 swap operations ?  
戊、最壞的情況下，需要請個 swap operations ?  
己、在此一問題中，如果只寫答案，且答案正確的話，可有 1/3 分數。  
庚、理由解釋正確且答案正確，可有全部分數。  
辛、理由解釋錯誤，即使答案正確，也不給分。  
壬、static void sort(Elem[] array)  
for (int i=1; i<array.length; i++)  
for (int j=i; (j>0) && (array[j].key()<array[j-1].key())); j--)  
swap(array, j, j-1);

- 九、(10%) 如果我們依序把下列資料：(10, 15, 19, 21, 24, 31, 45, 47, 50, 52, 12, 20, 30, 48, 18, 33, 23, 43, 22)，依序插入一個空的 2-3 tree，最後這棵樹會是甚麼樣子？請畫出。接著，我們再依序刪除 23, 30, 33, 22 這四筆資料，請問它會變成甚麼樣子？請畫出。

- 十、(5%) 輸入 48962 時，以下程式的輸出為何？

```
import javax.swing.JOptionPane;
public class Five {
    public static void main( String args[] )
    {
        int number;
        String inputString;

        inputString = JOptionPane.showInputDialog("Enter five digit integer.");
        number = Integer.parseInt( inputString );
        int digit1, digit2, digit3, digit4, digit5;
        digit1 = number / 10000;
        digit2 = number % 10000 / 1000;
        digit3 = number % 10000 % 1000 / 100;
        digit4 = number % 10000 % 1000 % 100 / 10;
        digit5 = number % 10000 % 1000 % 100 % 10;

        String resultString = digit1 + "    " + digit2 + "    " +
            digit3 + "    " + digit4 + "    " + digit5;
        JOptionPane.showMessageDialog( null, resultString,
            "Digits in " + number, JOptionPane.INFORMATION_MESSAGE );
        System.exit( 0 );
    } // end method main
} // end class Five
```

- 十一、(5%) 試繪出以下程式的結果 (並請標示出圓心及半徑)

```
import java.awt.Graphics;
import javax.swing.JApplet;
public class Circles extends JApplet {
    public void paint( Graphics g )
    {
```

```
super.paint( g );
int counter = 1;
do {
    g.drawOval( 110 - counter * 10, 110 - counter * 10,
               counter * 20, counter * 20 );
    ++counter;
} while ( counter <= 10 );
} // end method paint
} // end class Circles
```

十二、 (3%)試問底下程式的操作方式：

(4%) 執行時的用戶介面為何？

(3%)試說明 flip()內程式碼的意義。

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Coin extends JApplet implements ActionListener {
    JButton button;
    int heads, tails;

    public void init()
    {
        heads = 0;
        tails = 0;

        button = new JButton( "Toss" );
        button.addActionListener( this );
        Container container = getContentPane();
        container.add( button );
    }

    public void actionPerformed((ActionEvent e)
    {
        if ( flip() == true )
            heads++;
        else
            tails++;
        showStatus( "Heads: " + heads + " Tails: " + tails );
    }

    public boolean flip()
    {
        if ( (int)( Math.random() * 2 ) == 1 )
            return true;
        else
            return false;
    }
} // end class Coin
```

十三、 (5%)試問在 inputKey 物件輸入 "the" ; inputArea 物件輸入 "Knowledge work systems have characteristics that reflect the special needs of knowledge workers."後，程式的輸出結果為何？

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Index extends JFrame {
    private JTextField inputKey;
```

注意：背面有試題

```
private JLabel promptArea, promptKey;
private JTextArea inputArea;

public Index()
{
    super( "Puzzle" );
    inputKey = new JTextField( 4 );
    inputKey.addActionListener(

        new ActionListener() { // anonymous inner class
            public void actionPerformed( ActionEvent event )
            {
                int count = 0, current = 0;
                String inputKey = event.getActionCommand().toString();
                char key = inputKey.charAt( 0 );
                String inputString = inputArea.getText();
                current = inputString.indexOf( key, 0 );
                while ( current != -1 ) {
                    count++;
                    current = inputString.indexOf( key, current + 1 );
                }
                JOptionPane.showMessageDialog( null,
                    "key: " + key + "\ncount: " + count, "Results",
                    JOptionPane.INFORMATION_MESSAGE );
            }
        } // end anonymous inner class
    ); // end call to addActionListener

    promptArea = new JLabel( "Area" );
    promptKey = new JLabel( "key" );
    inputArea = new JTextArea( 4, 20 );
    JScrollPane scrollPane = new JScrollPane( inputArea );
    Container container = getContentPane();
    container.setLayout( new FlowLayout() );
    container.add( promptArea );
    container.add( scrollPane );
    container.add( promptKey );
    container.add( inputKey );
    setSize( 450, 150 );
    setVisible( true );
} // end constructor

public static void main( String args[] )
{
    Index application = new Index();
    application.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
}
} // end class Index
```