一．請在下列各小題內，選擇正確、最接近或最符合答案之選項號碼A、B、C、D之答案。（注意事項：作答時必須以適當間隔，依序標記每小題之題號，於題號之後寫填答案。答題所塗之選項號碼，請同試題用大寫A、B、C、D，未依注意事項作答造成答案無法確認，將予扣分或不予計分。）(每小題2%)

1.1 Most DBMSs support the creation of views. Which one of the following is incorrect? (A) Because each user has his or her own view, different users can view the same data in different ways. (B) Views can not provide data independence. (C) Views provide a measure of security. (D) A view is an application program’s or an individual user’s picture of the database.

1.2 Referring to functional dependence in database, which one of the following is incorrect? (A) One can determine functional dependence by looking at sample data. (B) Assume the values for column A in the relation are unique. Thus, all the other columns in the relation are functionally dependent on A. (C) Understanding functional dependence is crucial to learning normalization. (D) A column B is functionally dependent on another column A if each value for A in the database is related to exactly one value of B.

1.3 To implement a relationship in the ER (Entity Relationship) model, _______ of one entity appears as a foreign key in the related entity. (A) the derived attribute (B) the primary key (C) the secondary key (D) the related participant

1.4 Which one of the following is the advantage of the database with controlled redundancy? (A) Consistency (B) Independence (C) Integrity (D) Encapsulation

1.5 In a rollback, also called backward recovery, the DBMS uses the log to _______ any changes made to the database during a certain period. (A) backup (B) reenter (C) undo (D) modify

1.6 In the design of a relational database, the normalization process enables you to identify the existence of what potential problems? (A) concatenation (B) illegal determinants (C) incorrect decomposition (D) update anomalies

1.7 Which one of the following is a process-centered technique that is used to model business requirements for a system? (A) Structured analysis (B) Prototyping (C) Information engineering (D) object oriented analysis

1.8 The design strategy of rapid application development (RAD) calls for the interactive use of _______ and _______ to define the users' requirements and design the final system. (A) 3rd GL and 4GL (B) structured techniques and prototyping (C) UML and prototyping (D) DFD and ERD

1.9 Which one of the following is incorrect referring to object-oriented analysis and design? (A) Object technologies are an attempt to eliminate the separation of concerns about data and process. (B) Object modeling is a technique for identifying objects within the systems environment and the relationships
between those objects. (C) The UML prescribe a method for developing systems. (D) The object-oriented approach to system development is based on the concept that objects exist within a system’s environment.

1.10 What is the technique wherein the attributes and behaviors that are common to several types of object classes are grouped into their own class? (A) Encapsulation (B) Object modeling (C) Aggregation (D) Generalization

1.11 Which one of the following is a model-based technique wherein standard parameters based on prior projects are applied to the new project to estimate duration of a project and its tasks? (A) GANTT (B) PERT (C) COCOMO (D) WBS

1.12 Referring to physical data flow diagram (physical DFD) in systems analysis and design, which one of the following is incorrect? (A) In structured analysis and design, physical DFDs of the target system are intended to propose and model technology choices and design decisions for all logical processes, data flows, and data stores. (B) Physical DFDs serve as a technical blueprint for system construction and implementation. (C) Physical DFD is the primary tool used to develop the application architecture for the information system. (D) Systems analysts draw physical DFD to model the system’s raw data before they draw the data flow diagrams that illustrate how that data will be captured, stored, used, and maintained.

二、試分別為下列各項提出一樣應用，並說明理由。回答請簡單扼要。（每小題 2%）
2.1 Decision trees
2.2 K-means algorithms
2.3 Binary trees
2.4 Hash maps
2.5 Switched virtual circuit (SVC)
2.6 Sliding window
2.7 Mesh topology
2.8 Acknowledgement (ACK)
2.9 Cloud computing
2.10 Petri net
2.11 Minimum-cost spanning trees
2.12 Divide-and-conquer algorithms
2.13 Priority queues

三、為何電腦的記憶體空間要區分為核心記憶體 (kernel memory) 和使用者記憶體 (user memory)？(5%)
CPU要如何在這兩種空間中切換？(5%)

四、何謂虛擬機器 (virtual machine)？(5%)
虛擬機器可以分成哪些種類？(5%)
為何近年來虛擬機器技術在資訊界又受到很大的重視？(5%)

注意：背面有試題
五、某銀行運用Java及物件導向程式設計開發一個帳號管理(Account Management)系統(簡稱為AM系統)。當銀行收到客戶存入的交易(Transact)後會儲存(Deposit)款項，並記錄(Withdraw)款項。由於某種原因，系統將交易物件化，當客戶進行上述交易時，系統會根據所指定的交易類別產生(即具體化)一個相關的交易物件，客戶可繼續操作上述的交易功能，此時即連續產生多個交易物件，為了方便查考，這些交易物件會保留在系統中。系統並有一計數器來記錄目前剩餘的交易物件的數量。以下為上述的操作模擬所分析出的簡化之類別圖(class diagram)。

程式設計理念
1. 運用 static variable來設計上述的交易物件計數器
2. 在Demo类中針對Transaction及其子類別及accountBalance方法運用polymorphism及dynamic method binding
3. Transaction與Deposit及Withdraw事件繼承概念，以達成polymorphism的設計

請依上述的情境範圍、類別圖，和設計理念，完成下面的十道程式碼，並在答
案卷上依編號，工整地列出這十道程式答案。
注意：程式中的各種宣告已足夠使用。為能獲取本題所欲測試的目標，請勿再
重複宣告其他類別、方法或變數，否則將影響分數的給定。

```java
public class Account{
    private int balance; //帳號存款餘額，
    private String name; //存戶姓名
    private String AccNO; //帳號

    public Account(int balance, String name, String AccNO){
        (1) 完成本建構式(constructor) (3%)
    }

    public String getName(){
        return name; }
    public void setBalance(int balance){
        this.balance = balance; }
    public int getBalance(){
```
```java
public class Deposit extends Transaction{
    private Account a;
    public Deposit(Account a){
        super(a);
        this.a = a;
    }
    public void amendBalance(int amount){
        // 本方法是用來改變帳號的金額
        // (6) 完成本方法(amendBalance)的內容 (1.5%)
    }
}

public class Withdraw extends Transaction{
    private Account a;
    public Withdraw(Account a){
        super(a);
        this.a = a;
    }
    public void amendBalance(int amount){
        // (7) 完成本方法(amendBalance)的內容 (1.5%)
    }
}
```

```
return balance;
}

public (2) {

(2) 完成Transaction類別頭的部分 (2%)

private Account a;

(3) 在此宣告一個static variable "counter" (2%)

public Transaction(Account a){
    this.a = a;

(4) 完成本(Transaction)建構程式 (2%)
}

(5) 完成本(Transaction)類別 (3%)
}
```

```
注意: 背面有試題
```
import javax.swing.*;

public class Demo {
    public static void main (String args []){
        Account a = new Account(1000, "john", "8923345");
        Transaction t = null;
        int r1, r2;

        do {
            r1 = Integer.parseInt(JOptionPane.showInputDialog("請輸入交易型態: (1) deposit, (2) withdraw");

            switch (r1) {
                case 1:
                    t = new DepositAccount(a);
                    t.processTransaction();
                    break;
                case 2:
                    t = new WithdrawAccount(a);
                    t.processTransaction();
                    break;
                default:
                    JOptionPane.showMessageDialog(null, "Invalid option.");
                    break;
            }
        }

        JOptionPane.showMessageDialog(null, "您目前餘額: "+a.getBalance()
                + 
                "Number of transaction objects created: " + t.getNumTransactions());
    }
}