# **Chapter 1 Lecture Terms / Questions**

Use Chapter 1 of your textbook to find each of these concepts and add notes about them. Bring a PRINTED copy of your notes to class to use on the notes quiz.

1. Database
   1. A **database** is a collection of **information**
2. Examples of databases
   1. Your friendly bank keeps all your financial records on their **database**. When you receive your monthly statement, the bank generates a **database** **report**.
3. Database Management System (DBMS)
   1. You can surely think of many more places that databases enter your life. The idea is that they are everywhere. And, each database requires some way for a user to interact with the information within. Such interaction is performed by a **database** **management** **system** (**DBMS**).
4. DBMS tasks
   1. View the data
   2. Find some data of interest
   3. Modify the data
   4. Add some data
   5. Delete some data
5. 2 Commercial DBMS’s
   1. There are many commercial database management systems that perform these tasks. Programs like Access (a Microsoft product) and Oracle are used world-wide.
6. Visual C# as front end to the database
   1. Implementing a new application that requires management of a database
   2. Connecting to an existing database
   3. Interacting with a database via a server or the internet
7. Objects between C# and the database
   1. The **data** **objects** are Visual C# components that allow connection to the database, creation of data sets from the database and management of the database contents.
8. Why are there objects between them?
   1. These objects are the conduit between the application and the database, passing information back and forth between the two.
9. What do they do?
10. 2 Benefits of using C# as a DBMS
    1. Your users don’t need to have any commercial product installed on their computers or know how to use such products. This saves the users money.
    2. By building a custom front-end, you limit what your user can do with the information within the database. Under normal operation, commercial DBMS provide no such limits.
11. Forms (in C#)
    1. Windows that you create for user interface
12. Controls
    1. Graphical features drawn on forms to allow user interaction (text boxes, labels, scroll bars, buttons, etc.) (Forms and Controls are **objects**.)
13. Properties
    1. Every characteristic of a form or control is specified by a property. Example properties include names, captions, size, color, position, and contents. Visual C# applies default properties. You can change properties when designing the application or even when an application is executing.
14. Methods
    1. Built-in methods that can be invoked to impart some action to a particular control or object
15. Event Methods
    1. **Code** related to some object or control. This is the code that is executed when a certain event occurs. In our applications, this code will be written in the C# language (covered in detail in Chapter 2 of these notes).
16. General Methods
    1. **Code** not related to objects. This code must be invoked or called in the application.
17. 3 steps to build a Visual C# application
    1. **Draw** the user **interface**
    2. **Assign** **properties** to controls
    3. **Write** **code** for event methods. Develop any needed general methods.
18. Toolbox
    1. The **Toolbox** is the selection menu for controls (objects) used in your application.
19. Properties Window
    1. Properties button
       1. Its primary purpose is to establish design mode (initial) property values for objects (controls).
    2. Events button
       1. It can also be used to establish event methods for controls. Here, we just look at how to work with properties. To do this, click the **Properties** button in the task bar:
20. Solution Explorer Window
    1. The **Solution Explorer Window** displays a list of all forms and other files making up your application
21. Correct naming of controls
    1. This convention is to use a three letter (lower case) prefix (identifying the type of control) followed by a name you assign. A few of the prefixes are (we’ll see more as we progress in the notes):
22. Rules for naming any control
    1. Control (object) names can be up to 40 characters long, must start with a letter, must contain only letters, numbers, and the underscore (\_) character. Names are used in setting properties at run-time and also in establishing method names for control events. Use meaningful names that help you (or another programmer) understand the type and purpose of the respective controls.
23. Example of setting a property at run time
    1. objectName.PropertyName = NewValue;
24. Format for a control’s method
    1. Such a format is referred to as dot notation. For example, to change the **BackColor** property of a button named **btnStart**, we'd type:
    2. **btnStart.BackColor = Color.Blue;**
25. 2 ways to create an event for a control (like a button)
    1. To access a control’s default event method, you simply double-click the control on the form.
    2. To establish an event method using the properties window, click on the **Events** button (appears as a lightning bolt) in the properties window toolbar:
26. Rules for creating a variable name
    1. No more than 40 characters
    2. They may include letters, numbers, and underscore (\_)
    3. The first character must be a letter which, by convention, is usually lower case
    4. You cannot use a reserved word (word needed by Visual C#)
27. Different data types
    1. **bool** (true or false)
    2. **int**, **long** (Whole numbers)
    3. **short, float**, **double** (Floating point numbers)
    4. **DateTime**
    5. **string** (Used for many control properties)
    6. **char** (single character string variables)
    7. **Object** (yes, objects can be variables!)
28. Block level variable
    1. variables are only usable within a single block of code (will be discussed in more detail in Class 2).
29. Method level variable
    1. The value of **method** **level** variables are only available within a method. Such variables are declared within a method, using the variable type as a declarer:
30. Form level variable (and where are they declared)
    1. **level** variables retain their value and are available to all methods within that form. Form level variables are declared in the code window right after the **Form constructor** generated automatically by Visual C#, outside of any other method: