Applications Programming

Data Collection, Storage and Manipulation

Data Storage Design

- There must be meta data (field names) to help data interpretation.
- Break down your data as much as possible. It's easier to assemble data pieces together to interpret its whole meaning than to parse a piece of data to interpret its sub-meanings.
- All details logically related should be stored together.
- Independent facts should be stored separately.
- Avoid storing redundant (calculated) data.
- Always include an ID field, naturally or artificially, for any data collection if possible.

Data Integrity

- Inconsistency in a data collection would render the whole data collection un-trustworthy, therefore unusable, at all.
- Solutions to prevent data from being corrupted:
  - Apply good data storage design
  - Prevent inconsistent data entering the data collection: apply reasonable input validation rules when collecting the raw data
  - Detect and remove inconsistent data from the collection: review and apply further validation to clean the data
  - Use better data storage and manipulation tools (such as databases instead of Excel)

.

Database Systems

- Pseudo relational database system in Microsoft Office: Access
- Access vs Excel
    - Tables – Worksheets
    - Rows – Rows
    - Attributes – Columns
- Access over Excel
    - Efficient
    - Abstract in data storage and access
    - Build-in integrity control
    - Versatile security options

Accessing Data in Databases

- SQL query language
    - Easy for beginners
    - Declarative
    - Powerful enough
- Syntax
    - Select – pick your columns
    - From – choose your data source
    - Where – retrieve only the qualified rows/data objects
    - Order by – sort the result

Files

- A file is a collection of data stored on a computer disk. Data can be saved in a file and later reused.

- The following steps must be taken when a file is used by a program.

  - The file must be opened. Opening a file establishes a connection between the program and the file.

  - Data is written to the file or read from the file through the connection established in the previous step.

  - When the program is finished using the file, the file should be closed.

Open/Close a Text File

- To open a text file, first we need to know the file's name, including its path name if necessary

```
Dim filename As String
filename = "C:\Users\Documents\abc.txt"
```

- Then we can open the file using the following statement template:

```
Open filename For mode As #File_Descriptor
```

Where filename should be able to uniquely identify the file we want to open; mode indicates whether we want to open the file to read (Input) or to write (Output); File_Descriptor is a unique integer number that is assigned to this connection.

- For example:

```
' open abc.txt to read
Open filename For Input As #1
' then open def.txt in the same folder to write
Open "C:\Users\Documents\def.txt" For Output As #2
```

- To close a file, use

```
Close #fileDescriptor
```

Read/Write Text File

- •To read a line of data from a file (again, through a connection), using a statement in the following format:

  Line Input #fileDescriptor, buffer

  Where buffer must be a string variable to store the data read in from the file.

- To write data into a file (through a connection identified by the file descriptor, using a statement in the following format:

  Write #fileDesciptor, data

Excel Files (Workbooks)

- To open an Excel file named "abc.xlsx":

```
Dim W As Workbook
set W = Workbooks.Open("abc.xlsx")
```

- To read data from/write data to a cell in the workbook, we need to identify the worksheet, row and column of the cell. For example:

```
' read data from the cell "D3" in the first worksheet into variable x
x = W.Worksheets(1).Cells(3, 4)
' write a literal string into the cell "A1" in the worksheet named StudentInfo
W.Worksheets("StudentInfo").Cells(1, 1) = "Student Name“
```

- To close the workbook without saving the changes made by the program:

```
W.Close SaveChanges:=False
```

- To close the workbook and save the changes made by the program:

```
W.Close SaveChanges:=True
```