Computer Science – Lecture 20
Delphi File Handling

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk
Blackboard IFP

Objectives

- To talk about files
  - Reminder about Delphi I/O (input / output)
  - Things we want to do with files
  - How Delphi works with text files
  - Types of files

- Today's practical:
  - Writing data to a file

Delphi Input / Output

- We have used Delphi functions to read and write "string" data
  - Read
  - Readln
  - Write
  - Writeln

- These take input from the keyboard
- They write data to the "console" window

Writing Files

- Writing data to a file is similar to writing data to the console window

- BUT

- We need to identify which file to write to
- We might want to write to several files
- We cannot write and read from the same file
- What should happen if the file already has data in?

Reading Files

- Reading from a file is similar to reading from the keyboard

- BUT

- We need to identify the file
- We might want to read several files
- We cannot read and write from the same file
- We need to know when we get to the end of the file
- We don't have to read the file in order

Identifying Files

- To use a file we need a new variable type
- We *associate* a file name with this variable

```
Var

MyFile: TextFile;

Begin

AssignFile(MyFile, "Lotto.txt");
```

Writing Data to our file

- We must first decide what to do if the file already contains data
  - We could erase all the data and start again

```
rewrite(MyFile);
```

  - We could add our data to the end of the file

```
append(MyFile);
```

- We can then write data as required

```
write ( MyFile, "Some data" );
writeln ( MyFile, " more data" );
```

More Than One File

- We can write to as many files as we need
- Each file needs a TextFile variable
  - With a unique name!
- Use the appropriate variable in the write statement

Closing Files

- When we have finished writing data to a file we should close it
  - This will ensure that all the data is written to the disk
  - Other programs can now open the file
  - (E.g. you will not be able to open a file with notepad until your program closes that file)
- All files are automatically closed when your program finishes

Reading Data From Files

- To indicate that we will read data from the file we "reset" the file variable
  - `reset ( MyFile );`
- We can then read data as required
  - `readln ( MyFile, MyString );`
- (Remember, we cannot read and write the same file at the same time)
- If we want to read the same file again in the program we can call reset without closing and reopening the file

Running Out of Data to Read

- How do we know when we reach the end of the file?
  - Or if the file we are reading is empty?
- We use the End Of File function

```
EOF ( MyFile )
```

- This returns a boolean value
  - TRUE if we are at the end of the file
  - FALSE if there is still data in the file
- We can use this value in any of control loop statements, e.g.

```
until ( EOF ( MyFile ));
```

Types of Files

- So far we have been talking about Text Files
- These contain ordinary characters
  - Can be created and read by Notepad etc.

- There are other types of file
  - Often known as “data” or “binary” files
  - They cannot be read by Notepad (or display garbage)
  - For example, database files
- They can also be used from Delphi
  - But we will not be covering their use in this course

Summary

- File handling is similar to reading from the keyboard and writing to the console window

- But there are some differences
  - We need a file variable to work with
  - We have new functions to indicate if we are going to read or write the file, or if we want to append data
  - We can read or write several files
  - But not read and write the same file at the same time
  - We need a function to tell us if we are at the end of the file
  - And a means to close a file when we are finished with it

Today's Practical

- In previous practicals you produced a program to choose lottery numbers
  - If yours did not work fully, borrow some code from a fellow student!
- Modify the program to write the chosen numbers to a text file
  - The file should be called "lotto.txt"
  - You decide if your program will append or delete data
  - Write a title line in the file,
    - e.g. "Your winning numbers are"
  - Write each number on a separate line of the file
  - Run the program, then open your file in notepad