# Computer Data Types

## Basics of Computing

# Data Types

- Character
- Byte
- Boolean
- Integer
- Double
- Long
- String

# Character Data Type

- Stores 1 Character
- “Size” dependant upon language
- ASCII or EBCDIC Storage
- Examples
  - A
  - 2
  - “
  - -

# ASCII Character Set

- ASCII stands for American Standard Code for Information Interchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort.

# ASCII Character Set

- **ASCII was developed a long time ago and now the non-printing characters are rarely used for their original purpose.**
- **ASCII was actually designed for use with teletypes and so the descriptions are somewhat obscure.**

# ASCII Character Set

- Notepad.exe creates ASCII text, or in MS Word you can save a file as 'text only'

# EBCDIC Character Set

- EBCDIC (Extended Binary Coded Decimal Interchange Code) is a character encoding set used by IBM mainframes.
- IBM mainframes and midrange systems such as the AS/400 tend to use a wholly incompatible character set primarily designed for ease of use on punched cards.

# Why is EBCDIC Better than ASCII?

- EBCDIC is easier to use on punched cards
- Included the "cent sign" (¢) character that ASCII does not.

# Why ASCII Better Than EBCDIC?

- EBCDIC is a mess. The lack of contiguous character blocks make coding a real pain.
- Most of the world runs on ASCII. Even in IBM mainframe environments, host PCs, terminals and printers may use ASCII as their native character set.

# Why Is ASCII Better Than EBCDIC?

- Standard versions of EBCDIC miss out the ASCII characters **[ ] \ { } ^ ~ ¦** and include the **¢** sign, so there isn't even a direct match between them.

# Why Is ASCII Better Than EBCDIC?

- Worse still, some of the missing ASCII characters are in the UUencoding range, so it will tend to corrupt standard Internet mail attachments.

# Byte Data Type

- 8 Bits
- Basically 8 position binary number

# Boolean Data Type

- Yes/No
- True/False
- A/B
- Size dependant upon language

# Integer Data Type

- -65536 to 65536
- Size usually 2 bytes
- INTEGER only
  - No decimals
  - No Fractions

# Double Data Type

- -1.79769313486231570E+308 through -4.94065645841246544E-324 for negative values and from 4.94065645841246544E-324 through 1.79769313486231570E+308 for positive values.
- Size - usually 8 bytes

# Long Data Type

- -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807
- Integer Values Only
- Size - Usually 8 bytes

# String Data Type

- Groups of Characters
- Usually Specific Size Defined by Programmer
- Examples
  - APPLE
  - SMITH
  - 888-33-9999