# Strings

The storage of text items

# Why not use characters?

- Characters are only one character long
- Other forms of text will need to be catered for
  - Names
  - Addresses
- Also, File I/O items are taken in as strings (which we will see later in the course)
- Need to know how to manipulate strings

# Strings are Objects

- Not a Primitive type
- String is an OBJECT
  - Denoted by capital S
- String name = “Ian”;
    - This is one way to declare a string – will see others later
- **Can call up extra functionality**
    - By use of the . Operator (dot operator)
      - .length()
      - .substring()

# Main String functions

- length:
  - the length of the string, including spaces
- indexOf:
  - returns the position of a character or string
- charAt:
  - returns what character is at a given position
- substring
  - returns a smaller string within the string
- touppercase/tolowercase
  - Converts case of string

# Creating and Initialising Strings

As stated before, several means of declaring and initialising strings

- String x = new String();
  - Further in the code we then assign a value to x
- String message = new String("Welcome to Java");
  - Creating a string called message and assigning a value immediately
- Since strings are used frequently, Java provides a shorthand initializer for creating a string:
  - String message = "Welcome to Java";

# Strings as IMMUTABLE objects

- An immutable object is one that CANNOT be changed
  - In the case of Strings, it means that we have to reassign the new value to the original string if we want to change it
- EXAMPLE:
  - The concat() function within String concatentes the existing string with new text i.e. Adds to it

1) String x = “Hello”;
2) x.concat(“ world”);
3) x = x.concat(“ world”);

# Immutability (Concat example)

- Only one of the lines 2 and 3 will construct the phrase “Hello World”
- Line 3 is the correct one
  - It performs the concatenation and then assigns the value to the string x, replacing its existing value
- So what happens at line 2 in relation to x?
  - NOTHING!
  - It does actually create a new unreferenced object but since we don't know its name, it is lost

# Immutable Objects

- It is always good to know what objects are mutable and immutable
  - If immutable, you need to distinctly assign it to some variable in order to see the variable again
  - If not (if the object is mutable), any function call will take effect
    - StringBuilder, which we will look at later in the course, is a mutable class

# Links

1. http://www.tutorialspoint.com/java/java_strings.htm - Summary of Strings
2. http://www.javatpoint.com/methods-of-string-class - Good summary of important functions