Lesson 2 Handout v0.1

1 Terminology

1.1 **string** – A data type that contains text. Essentially, this is a collection of individual characters that are ‘strung’ together much like beads on a necklace. Each character is actually a number that represents a character, but special references must be used to treat an individual character as its number.

1.2 **concatenation** – ‘Adding’ two or more strings together to make a new string composed of all of them.

1.3 **slicing** – A technique that can be used on a collection of data items, such as the characters in a string. Sections of a string can be referenced based on a starting and ending position.

1.4 **white space** – Often means the blank space in a string.

1.5 **dot-operator** – When a period is used between twoe labels, it is a dot-operator.

1.6 **object oriented programming (OOP)** – A type of programming that always for labels to have a context sensitivity, among other things. OOP will be covered later.

1.7 **nested function call** – When a function call is included as an argument to another function call.

1.8 **data-type-conversion** – When the data of one type is converted to another type.

1.9 **exception** – A kind of error that can occur during the execution of a program.

1.10 **input-validation** – A technique in programming where data is evaluated to ensure it is of appropriate type and value. This will be covered later.

2 Programming Elements

2.1 Operators:

2.1.1 . - The ‘dot’-operator. Used frequently in Object Oriented Programming.

2.1.2 **+ - * /** % - mathematical operators (note that *, /, and % are multiply, divide, and modulus respectively)

2.1.3 ***string_variable* + *string_variable*** – Two or more string variables may be concatenated using the addition operator.

2.1.4 ***string_variable*[*n*]** – Square brackets which immediately proceed a string variable can contain the position of a single character to reference from within it. Note that referencing a position outside of the string will result in an error or exception.

2.1.5 ***string_variable*[*n1*:*n2*]** – A ‘slice’ or range of characters within a string can be referenced from within a string. Note that referencing a position outside of a string will result in an error or an exception.

Clueless to Coding – Lesson 2 Handout

2.2 Python Keywords:

This section left blank intentionally.

2.3 Python Functions:

2.3.1 **len(*variable*)** - This function will return an integer representing the 'length' of a variable. For strings this will normally be the number of characters it contains.

2.3.2 **str.strip(*string*)** – This function 'strips' white space from the front and back of a string.

2.3.3 **str.lower(*string*)** – This function changes all letters within a string to lower-case.

2.3.4 **str.upper(*string*)** – This function changes all letters within a string to upper-case.

2.3.5 **str.capitalize(*string*)** – This function capitalizes the first character within a string, and make all other alphabetical characters lower-case.

2.3.6 **str.title(*string*)** – This function finds each word within a string and capitalizes it.

2.3.7 **input(*[optional prompt string]*)** - This function will collect input from a user and return it as a text string. An optional string may be included as an argument, which will be used as a prompt.

2.3.8 **str(*argument*)** – Returns a string version of the argument.

2.3.9 **int(*argument*)** – Attempts to find an integer value and returns it. Will throw an exception if the attempt fails; this typically happens when a string argument passed to it does not contain a number.

2.3.10 **float(*argument*)** – Attempts to find a floating point value and returns it. Will throw an exception if the attempt fails; this typically happens when a string argument passed to it does not contain a number.