**Validation**

Validation is the automatic checking of entered data by a computer program. Validation cannot check that the data entered is correct only that it is a reasonable value.

Complete the first column of this table using the words listed below, to match the validation type with the correct description. Then give an example of each type in the third column.

*presence range length type look-up*

|  |  |  |
| --- | --- | --- |
| **Type of validation** | **Description** | **Example** |
| Length | Checks the data entered is not too short or too long. |  |
| Presence | Checks that data has been entered. |  |
| Range | Checks that the value entered falls within a given range. |  |
| Type | Checks that the value entered is of the expected type. |  |
| Look up | Checks the entered value is a value that is expected. Checks value against a look up list or string. |  |

**Activity 16.2**

**Validation – length check**

Write a program that asks the user to input a password and then uses a length check to make sure the password is at least eight characters long. If it is shorter than eight characters the user is asked to enter a different password.

*Hint: Use the len() function.*

|  |
| --- |
| *Password = Input(“Enter a password.”)*  *Len(password)*  *while Password != <8 :*  *print("Enter another password.")*  *Password = input("Please respond.")*  *print("get something to eat!")*  *print(count)* |

**Activity 16.3**

**Validation – presence check**

Write a program that asks the user to enter a name and uses a presence check to make sure that an answer has been entered. If nothing has been entered the user is prompted again to enter a name.

|  |
| --- |
|  |

Extension: The program should output an appropriate message when the user fails to enter any characters.

|  |
| --- |
|  |

**Activity 16.4**

**Validation – type check**

Write a program that asks the user to enter their age and checks that they have entered an integer. It should display a message asking them to enter a number if they have not done so.

*Hint: Use the ‘try except else’ command.*

|  |
| --- |
|  |

**Activity 16.5**

**Validation – presence check**

Write a program that asks the user to enter an email address and then checks the string entered to make sure it contains an ‘@’. If it does not, the user is prompted again to enter an email address.

|  |
| --- |
|  |

**Activity 16.6**

**Validation – range check**

Write a program that asks the user to enter a % of charge left in their mobile phone. Use a range check to make sure the value is less 100% or more than 0%. The program must ask the user to re-enter the value if it is outside the range.

|  |
| --- |
|  |

Extension: Alter the program so it only allows integer numbers to be entered.

|  |
| --- |
|  |

**Activity 16.7**

**Try command: divide by zero error check**

Write a program that asks the user for two numbers, then divides the numbers and displays the answer. If the program generates a divide by zero error, display a message to explain they entered a zero as the second number.

|  |
| --- |
|  |

**Activity 16.8**

Write a program that asks the user to enter their name, age and email address using validation to ensure the data contains reasonable values. It should then display the data and ask the user if it is correct.

Use validation to make sure the user can reply ‘Y’, ‘y’ or ‘Yes’ and ‘N’, ‘n’, or ‘No’. The user should be allowed to re-enter the data if it is incorrect.

|  |
| --- |
|  |