```
############################## QS_Input_Validator ###################################

#### Author(s): Paul R Phillips, Bryan Green, Dylan Martin
#### Date Created: 1/14/18
#### Description

# The QS_Input_Validator prompts the user to input 3 values (a, b, c). This format follows the assumption
# the input values are derived from a standard quadratic equation (ax^2 + bx + c). After the values are
# inputted the function first stores these values as (a, b, c), then converts them to numeric format
# (Note: In R, the numeric format allows for calculation of both integer and double formats). Once values
# are converted the function's primary purpose is to indentify NaN's, NA's, +-Inf, character values,
# and if a = 0. Appropriate messages are displayed to user if any of these values are detected. If
# input values are in the appropriate format, the the message
# "Message: Input meets criteria for calculations." is returned to user (which is the desired outcome for
# this function).

#### Objects, attributes

# - a, b, c (character): Initial storage of inputted values
# - a_convert, b_convert, c_convert (numeric): a, b, c objects converted to R's numeric format
#   - Note: a, b, c are kept in function for user-interpretability purposes with respect to return messages
# - output (list): object returned to user
#   - Note: This object typically stores a message for user, as well as input values for users to see
# - Message (character): message returned to user; typically formatted as either an "Error" or "Warning".
# - Input (character or numeric): object used to return inputted values to user; if values pass all checks
#                                 it is returned in numeric format, otherwise in character

#### Return Values

# $Message
# [1] "Message: Input meets criteria for calculations."

# $Input
# [1] a_convert b_convert c_convert

#### Error codes: Return messages for NaN's, NA's, +-Inf's, and character values

# $Message
# [1] "Error: Non-numeric or infinite values have been detected.: <value>"

# $Input
# [1] a b c

#### File Path: C:\Users\paulp\Desktop\DataScience\WMU\CS4900\JKK_Consultant\QuadSolver_Input_Validation

#### OS specifications:

# Windows edition: Windows 10 Home
# System type: 64-bit Operating System, x64-based processor

#### R version: 3.4.3 (2017-11-30)

# Platform: Platform: x86_64-w64-mingw32/x64 (64-bit)

# Note: These conditional statements catch character values that were converted to NA's (since R converts
#       character values to NA's if present).
# Note: All values inputted are initially stored as character values. This is the protocol for the readline
#       function


######################################## End of Comment Block ##############################################

QS_Input_Validator <- function(x) {

# User-prompt for input: values 'a', 'b', and 'c'

   a <- readline("what is the value of 'a'? ")

   b <- readline("what is the value of 'b'? ")

   c <- readline("what is the value of 'c'? ")

# Conversion from character to numeric (IEEEfp double precision and integer format is implemented currently)

   a_convert <- as.numeric(a)

   b_convert <- as.numeric(b)

   c_convert <- as.numeric(c)

# Flags for incorrect input (specifically missing values, character values, and Inf values provided)
# Note: 'a', 'b', and 'c' are evaluated individually
# - if non-numeric or Inf values are present, an error message is returned to user,
#    along with the value that was converted to NA
# - NaN's are caught at this step, will return the error message if inputted

   if (is.na(a_convert) || is.infinite(a_convert)) {

      output <- paste0("Error: Non-numeric or infinite values have been detected.: ", a_convert)

      values <- c(a, b, c)

      output <- list(Message = output, Input = values)

      return(output)

   }

   else if (is.na(b_convert) || is.infinite(b_convert)) {

      output <- paste0("Error: Non-numeric or infinite values have been detected.: ", b_convert)

      values <- c(a, b, c)

      output <- list(Message = output, Input = values)

      return(output)
   }

   else if (is.na(c_convert) || is.infinite(c_convert)) {

      output <- paste0("Error: Non-numeric or infinite values have been detected.: ", c_convert)

      values <- c(a, b, c)

      output <- list(Message = output, Input = values)

      return(output)
   }

# Checking if a = 0
# - if a = 0 is TRUE, a message is returned indicating a cannot equal zero
# - if a = 0 is FALSE, a message is returned indicating input meets criteria, along with input values

   if(a != 0) {

      output <- paste("Message: Input meets criteria for calculations.")
```

```
      values <- c(a_convert, b_convert, c_convert)

      output <- list(Message = output, Input = values)

   }

   else {

      output <- paste0("Error: Value 'a' cannot equal zero")

      values <- c(a, b, c)

      output <- list(Message = output, Input = values)

      return(output)
   }

   return(output)

}
```