Problem-Solving Steps:

1. **Import the necessary libraries:** In this case, we import *cv2* for computer vision tasks and *numpy* for array manipulation.
2. **Define the function:** We define a function named *measure_object_dimensions* to encapsulate the code for measuring object dimensions.
3. **Capture video frames:** We open a video capture object using *cv2.VideoCapture(0)* to capture frames from the webcam.
4. **Process frames:** We enter a while loop to continuously read frames from the video capture object.
   - **Convert frames to grayscale:** We convert the color frames to grayscale using *cv2.cvtColor()*.
   - **Threshold frames:** We apply thresholding using *cv2.threshold()* to convert the grayscale frames to binary images.
   - **Find contours:** We find the contours of objects in the frame using *cv2.findContours()*.
   - **Find the largest contour:** We use *max()* and *cv2.contourArea()* to find the largest contour, representing the complete object.
   - **Get the bounding rectangle:** We extract the bounding rectangle of the largest contour using *cv2.boundingRect()*.
5. **Draw bounding rectangle and display dimensions:** We draw the bounding rectangle on the frame using *cv2.rectangle()* and display the width and height of the object using *cv2.putText()*.
6. **Display the frame:** We display the frame with the object dimensions using *cv2.imshow()*.
7. **Quit the program:** We check for the 'q' key or the Esc key using *cv2.waitKey()* and break the loop if either key is pressed.
8. **Release resources:** We release the video capture object and close any open windows using *cap.release()* and *cv2.destroyAllWindows()*.