Code	Annotations
import pandas as pd import numpy as np	This imports the pandas and numpy packages for our code.
cubic_yard_ft3 = 27 unit_weight_water = 62.4	This establishes constants and assigns them each a name.
print("Units:") print("Cement, Water, Aggregates Units: lb/yd^3") print("Specific Gravity: Dimensionless") print("Air Content: %") print("Target water-cement (w/c) ratio: Dimensionless\n")	This states the units of each variable.
print("Constants:") print(f"Defined constant: 1 cubic yard = {cubic_yard_ft3} cubic feet") print(f"Unit weight of water = {unit_weight_water} lb per cubic feet""\n")	This states the constants that were just defined.
print("Dictionary:") print("cement_A: weight of cement") print("fly_ash_B: weight of fly ash") print("silica_fume_C: weight of silica fume") print("other_scm_D: weight of other SCM") print("wc_ratio_E: target water-cement ratio") print("air_content_F: target air content (%)") print("percent_fine_G: percent of fine aggregate") print("percent_coarse_H: percent of coarse aggregate") print("percent_other_I: percent of other aggregate") print("sg_cement_J: specific gravity for cement") print("sg_fly_ash_K: specific gravity for fly ash") print("sg_silica_fume_L: specific gravity for silica fume") print("sg_other_scm_M: specific gravity for other SCM") print("sg_fine_N: specific gravity of fine aggregate") print("sg_coarse_O: specific gravity of coarse aggregate") print("sg_other_P: specific gravity of other aggregate\n")	This code creates a “Dictionary” and states all variables and what each of them signify.

def calculate_water_weight_Q(cement_A, fly_ash_B, silica_fume_C, other_scm_D, wc_ratio_E):
  total_cementitious = cement_A + fly_ash_B + silica_fume_C + other_scm_D
  water_weight_Q = total_cementitious * wc_ratio_E
  return water_weight_Q
	This defines a new variable that calculates the total weight of water given the inputs.

def calculate_volume_R(cement_A, sg_cement_J):
  volume_R = cement_A / (sg_cement_J * unit_weight_water)
  return volume_R
 	
def calculate_volume_S(fly_ash_B, sg_fly_ash_K):
  volume_S = fly_ash_B / (sg_fly_ash_K * unit_weight_water)
  return volume_S
 	
def calculate_volume_T(silica_fume_C, sg_silica_fume_L):
  volume_T = silica_fume_C / (sg_silica_fume_L * unit_weight_water)
  return volume_T
 	
def calculate_volume_U(other_scm_D, sg_other_scm_M):
  volume_U = other_scm_D / (sg_other_scm_M * unit_weight_water)
  return volume_U
	Each of these code sections defines a new variable to describe the volume of cement, fly ash, silica fume, and other SCM.

def calculate_air_volume_V(air_volume_F):
  volume_V = (air_content_F /100) * cubic_yard_ft3
  return volume_V
 	
def calculate_water_volume_W(water_weight_Q):
  volume_W = (water_weight_Q / unit_weight_water)
  return volume_W
	This code defines a new variable that calculates to volume of air, and then water, in the mix.

def calculate_total_aggregate_volume_X(volume_R, volume_S, volume_T, volume_U, volume_V, volume_W):
  volume_X = cubic_yard_ft3 - volume_R - volume_S - volume_T - volume_U - volume_V - volume_W
  return volume_X
	This code defines a new variable that calculates the total aggregate volume by subtracting all other volumes from 1 cubic yard.

def calculate_fine_aggregate_Y(percent_fine_G, sg_fine_N, volume_X):
  weight_Y = unit_weight_water * (percent_fine_G / 100) * sg_fine_N *volume_X
  return weight_Y
 	
def calculate_coarse_aggregate_Z(percent_coarse_H, sg_coarse_O, volume_X):
  weight_Z = unit_weight_water * (percent_coarse_H / 100) * sg_coarse_O * volume_X
	Each section of this code uses Volume X from the previous row to define a new variable to calculate the weight of fine, coarse, and other aggregates.

return weight_Z def calculate_other_aggregate_AA(percent_other_I, sg_other_P, volume_X): weight_AA = unit_weight_water * (percent_other_I / 100) * sg_other_P * volume_X return weight_AA	
project_no = int(input("Enter Project Number: ")) concrete_class = input("Enter Class of Concrete: ") cement_A = float(input("Enter cement content A (lb/yd^3): ")) fly_ash_B = float(input("Enter fly ash content B (lb/yd^3): ")) silica_fume_C = float(input("Enter silica fume content C (lb/yd^3): ")) other_scm_D = float(input("Enter other scm content D (lb/yd^3): ")) wc_ratio_E = float(input("Enter target water-cement ratio E: ")) air_content_F = float(input("Enter target air content F (%): ")) percent_fine_G = float(input("Enter percent fine aggregate G (%): ")) percent_coarse_H = float(input("Enter percent coarse aggregate H (%): ")) percent_other_I = float(input("Enter percent other aggregate I (%): ")) sg_cement_J = float(input("Enter specific gravity of cement J: ")) sg_fly_ash_K = float(input("Enter specific gravity of fly ash K: ")) sg_silica_fume_L = float(input("Enter specific gravity of silica fume L: ")) sg_other_scm_M = float(input("Enter specific gravity of other scm M: ")) sg_fine_N = float(input("Enter specific gravity of fine aggregate N: ")) sg_coarse_O = float(input("Enter specific gravity of coarse aggregate O: ")) sg_other_P = float(input("Enter specific gravity of other aggregate P: "))	This code makes the prompts of which the user will input all the variables so that the calculations may be done.

Q = calculate_water_weight_Q(cement_A, fly_ash_B, silica_fume_C, other_scm_D, wc_ratio_E) R = calculate_volume_R(cement_A, sg_cement_J) S = calculate_volume_S(fly_ash_B, sg_fly_ash_K) T = calculate_volume_T(silica_fume_C, sg_silica_fume_L) U = calculate_volume_U(other_scm_D, sg_other_scm_M) V = calculate_air_volume_V(air_content_F) W = calculate_water_volume_W(Q) X = calculate_total_aggregate_volume_X(R, S, T, U, V, W) Y = calculate_fine_aggregate_Y(percent_fine_G, sg_fine_N, X) Z = calculate_coarse_aggregate_Z(percent_coarse_H, sg_coarse_O, X) AA = calculate_other_aggregate_AA(percent_other_I, sg_other_P, X)	This code takes the previously defined volumes and assigns them a single letter to make the subsequent coding easier.
print(f"\nStep 1: Water Weight") print(f"Q (Water Weight) = {Q:.3f}") print(f"\nStep 2: Cementitious Volumes") print(f"R (Cement Volume) = {R:.3f}") print(f"S (Fly Ash Volume) = {S:.3f}") print(f"T (Silica Fume Volume) = {T:.3f}") print(f"U (Other SCM Volume) = {U:.3f}") print(f"\nStep 3: Air and Water Volumes") print(f"V (Air Volume) = {V:.3f}") print(f"W (Water Volume) = {W:.3f}") print(f"\nStep 4: Total Aggregate Volume") print(f"X (Aggregate Volume) = {X:.3f}") print(f"\nStep 5: Aggregate Weights") print(f"Y (Fine Aggregate) = {Y:.3f}") print(f"Z (Coarse Aggregate) = {Z:.3f}") print(f"AA (Other Aggregate) = {AA:.3f}")	This code prints all of the calculated variables as part of the output.
print("\n----------") print(" NDOT Concrete Mix Design – Weight Summary") print(" (1 Cubic Yard of Concrete)") print("----------") print(f"Project Number: {project_no}") print(f"Class of Concrete: {concrete_class}") print("----------")	This code is also part of the output, and summarizes the calculated output into a neat table.

print(f"Cement (A): {cement_A:8.1f} lb") print(f"Fly Ash (B): {fly_ash_B:8.1f} lb") print(f"Silica Fume (C): {silica_fume_C:8.1f} lb") print(f"Other SCM (D): {other_scm_D:8.1f} lb") print("----------") print(f"Fine Aggregate (Y): {Y:8.0f} lb") print(f"Coarse Aggregate (Z): {Z:8.0f} lb") print(f"Other Aggregate (AA): {AA:8.0f} lb") print("----------") print(f"Water (Q): {Q:8.0f} lb") print("----------") print("End of Mix Design Summary") print("\nAll mix design calculations completed successfully.")	
FOLLOWING CODE IS REPEATED FOR 4 DIFFERENT SCENARIOS	
print("Mix Design Summary Senario 1") print("Source: NDOT") print("Use Case: Main Highway Pavement, Baseline pavement mix") print("Inputs:") print(f"Project Number: {project_no}") print(f"Class of Concrete: {concrete_class}") print(f"cement content A (lb/yd^3): {cement_A}") print(f"fly ash content B (lb/yd^3): {fly_ash_B}") print(f"silica fume content C (lb/yd^3): {silica_fume_C}") print(f"other scm content D (lb/yd^3): {other_scm_D}") print(f"target water-cement ratio E: {wc_ratio_E}") print(f"target air content F (%): {air_content}") print(f"percent fine aggregate G (%): {percent_fine_G}") print(f"percent coarse aggregate H (%): {percent_coarse_H}") print(f"percent other aggregate I (%): {percent_other_I}") print(f"specific gravity of cement J: {sg_cement_J}") print(f"specific gravity of fly ash K: {sg_fly_ash_K}") print(f"specific gravity of silica fume L: {sg_silica_fume_L}") print(f"specific gravity of other scm M: {sg_other_scm_M}") print(f"specific gravity of fine aggregate N: {sg_fine_N}") print(f"specific gravity of coarse aggregate O: {sg_coarse_O}") print(f"specific gravity of other aggregate P: {sg_other_P}")	This code prints out a mix design scenario, including the source, uses, and everything that was inputed to accomplish the mix design.

print("-" * 55) print(f"{'Component':<20}{'Weight (lb/yd^3)':<20}{'Notes'}") print("-" * 55) print(f"{'Cement':<20}{cement_A:<20.1f}") print(f"{'Water':<20}{Q:<20.1f}") print(f"{'Fine Aggregate':<20}{Y:<20.1f}") print(f"{'Coarse Aggregate':<20}{Z:<20.1f}") print(f"{'Air (volumetric)':<20}{V:<20.1f}{'percent / ft^3'}") total_weight = cement_A + Q + Y + Z print("-" * 55) print(f"{'Total (sanity check)':<20}{total_weight:<20.3f}") print("-" * 55)	This code prints a summary of the scenario output in a neat table.
print(f"The mix design for Project {project_no}, Class {concrete_class} includes a cement weight of {cement_A:.1f}, water weight of {Q:.1f},") print(f"fine aggregate weight of {Y:.1f}, coarse aggregate weight {Z:.1f} (all in lb/yd^3), and air volume of ") print(f"{V:.1f} %/ft^3. The total weight was {total_weight:.1f} lbs.")	This code gives a short English summary of the scenario output for the client.