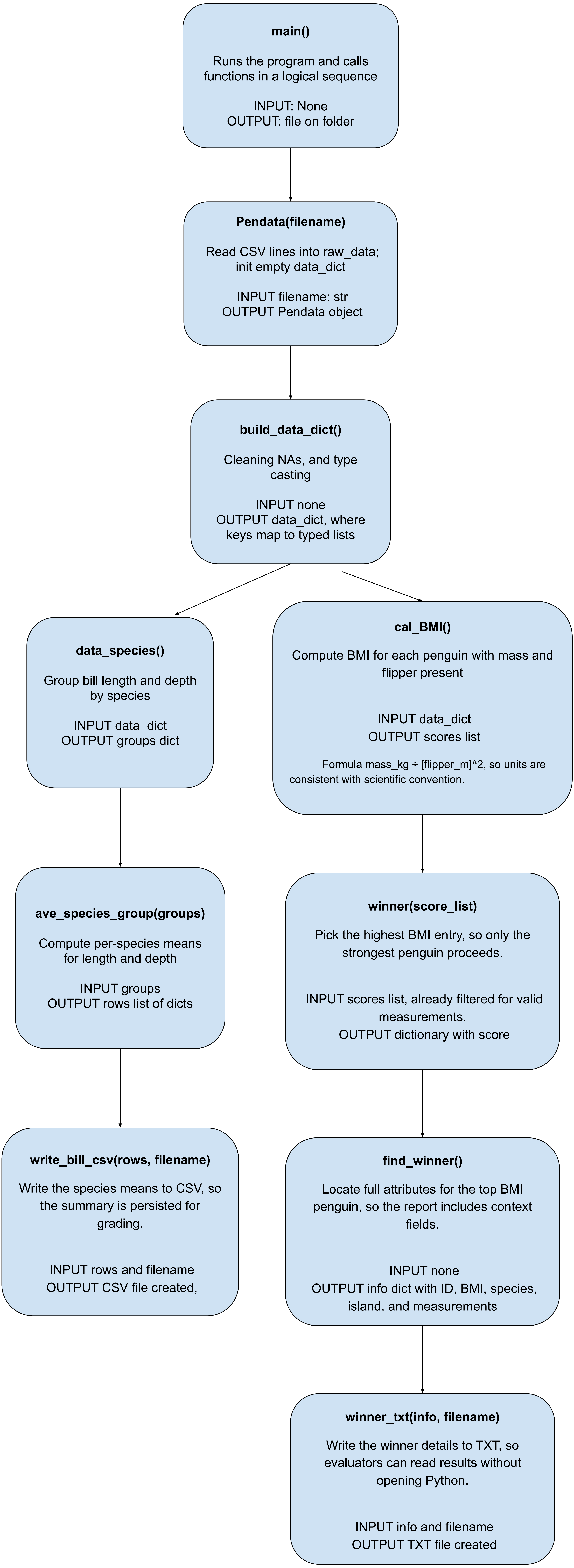

main()

Runs the program and calls functions in a logical sequence

INPUT: None
OUTPUT: file on folder

Pendata(filename)

Read CSV lines into raw_data; init empty data_dict

INPUT filename: str
OUTPUT Pendata object

build_data_dict()

Cleaning NAs, and type casting

INPUT none
OUTPUT data_dict, where keys map to typed lists

data_species()

Group bill length and depth by species

INPUT data_dict
OUTPUT groups dict

ave_species_group(groups)

Compute per-species means for length and depth

INPUT groups
OUTPUT rows list of dicts

write_bill_csv(rows, filename)

Write the species means to CSV, so the summary is persisted for grading.

INPUT rows and filename
OUTPUT CSV file created,

cal_BMI()

Compute BMI for each penguin with mass and flipper present

INPUT data_dict
OUTPUT scores list

Formula mass_kg ÷ [flipper_m]^2, so units are consistent with scientific convention.

winner(score_list)

Pick the highest BMI entry, so only the strongest penguin proceeds.

INPUT scores list, already filtered for valid measurements.
OUTPUT dictionary with score

find_winner()

Locate full attributes for the top BMI penguin, so the report includes context fields.

INPUT none
OUTPUT info dict with ID, BMI, species, island, and measurements

winner_txt(info, filename)

Write the winner details to TXT, so evaluators can read results without opening Python.

INPUT info and filename
OUTPUT TXT file created