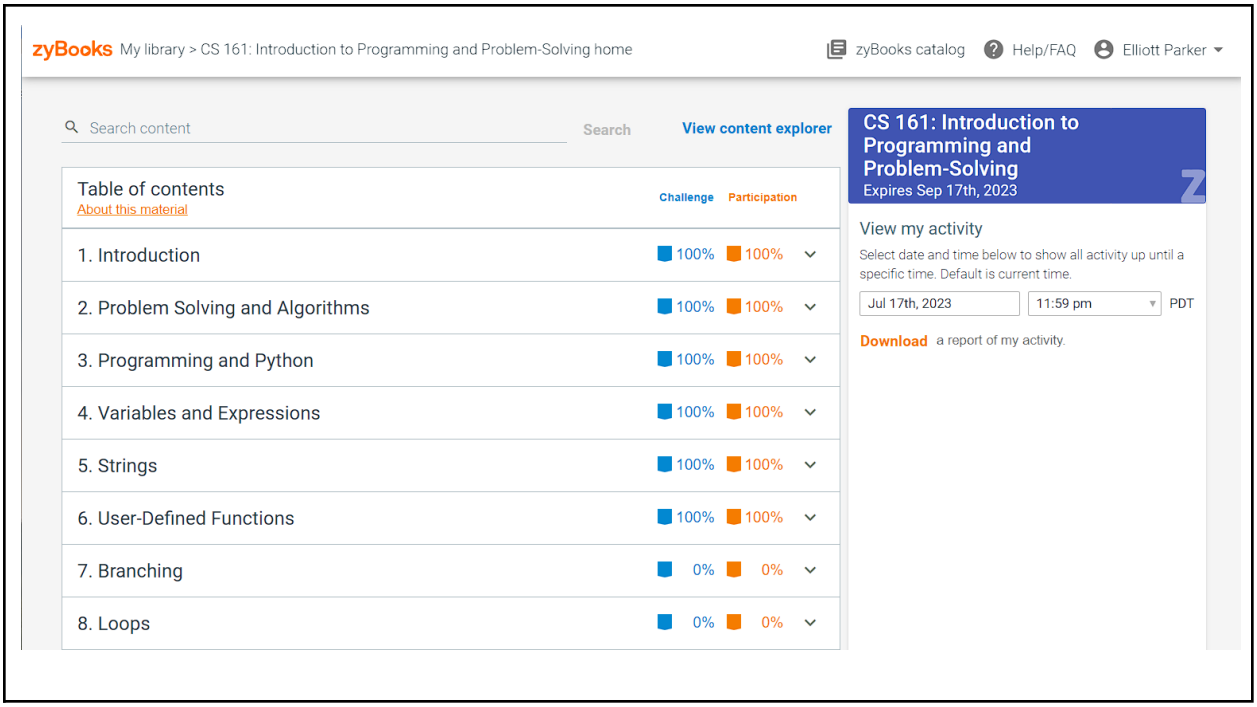
Algorithmic Planning Document

*Make a copy of this document to help you with your program planning (File → Make a copy). This document contains an interactive checklist. To mark an item as complete, click on the box. For **Assignments**, download the document as a PDF (File → Download → PDF) and submit to the assignment in Canvas.*

zyBooks Screenshot

When you are ready to upload your Python code to Canvas, take a screenshot of your zyBooks completion percentages for the assigned chapters and paste it in the box below. You must complete 70% of the zyBooks challenge and participation activities before your assignment will be graded.



Planning your program before you start coding is part of the development process. In this document, you will:

- ☑ ~~Step 1: Write a detailed description of your program~~
- ☑ ~~Step 2: Design a sample run with test input and output~~
- ☑ ~~Step 3: Algorithm design~~
  - ☑ ~~Identify the program inputs and their data types~~
  - ☑ ~~Identify the program outputs and their data types~~
  - ☑ ~~Identify any calculations or formulas needed~~
  - ☑ ~~Identify input/process/output functions and write the algorithmic steps~~

1. Program Description

In the box below, describe the purpose of the program. You must include a detailed description with at least two complete sentences.

Program Description:
This program takes the user's first name and location (as a zip code) as inputs. The first API converts the ZIP code to the Longitude and Latitude coordinates. Another API then pulls the temperature, wind speed, and humidity for their location. The program then calculates the heat index if the temperature is above 80 degrees Fahrenheit. If the temperature is below 50 degrees, the program will also pull the wind speed. The wind speed will be used to calculate the wind chill. The program is being refactored to use functions. Added several functions to store the results from the API responses.

2. Sample Run

If you are designing your program, you will start with a sample run. Imagine a user is running your program - what will they see? What inputs do you expect, and what will be the outputs from the given inputs? Choose the test data you will use to test your program. Calculate and show the expected outputs. Use the sample run to test your program.

Sample run:
Welcome to the Weather App! Please enter your name: Bob Please enter your Zip code: 90210 The current temperature in Beverly Hills is 81.2 degrees. And with the heat index, it feels like 85.9 degrees Fahrenheit. Have a great day, Bob!

3. Algorithmic Design

Before you begin coding, **you must first plan out the logic** and think about what data you will use to test your program for correctness. All programmers plan before coding - this saves a lot of time and frustration! Use the steps below to identify the inputs and outputs, calculations, and steps needed to solve the problem.

Algorithmic design:
a. Identify and list all of the user input and their data types. You must have at least three inputs, and one **must** be a string. I do not recommend having more than five inputs.

Name (String)

Location (Integer) #chaged to zip

API Key (String)

API Endpoint (String)

b. Identify and list all of the output and their data types.

Heat Index (Integer)

Wind Chill (Integer)

c. What calculations do you need to do to transform inputs into outputs? List all formulas needed, if applicable. If no calculations are needed, state there are no calculations for this algorithm.

Heat Index = -42.379 + 2.04901523*T + 10.14333127*RH - .22475541*T*RH - .00683783*T*T - .05481717*RH*RH + .00122874*T*T*RH + .00085282*T*RH*RH - .00000199*T*T*RH*RH

Wind Chill = 35.74 + 0.6215T – 35.75 (V^0.16) + 0.4275T (V^0.16)

d. Write out the steps the program will perform. Each step will be translated directly into Python code. **DO NOT use code syntax, think about the steps you would ask someone to follow to solve the problem in conversational words.**

main function steps:

1. Declare Constants
    a. Heat Index
        i. c_1 = -42.379,
        ii. c_2 = 2.04901523,
        iii. c_3 = 10.14333127,
        iv. c_4 = -0.22475541,
        v. c_5 = -6.83783e-3,
        vi. c_6 = -5.481717e-2,
        vii. c_7 = 1.22874e-3,
        viii. c_8 = 8.5282e-4,
        ix. c_9 = -1.99e-6
    b. Windchill

i. d_1 = 35.74
ii. d_2 = 0.6215
iii. d_3 = – 35.75
iv. d_4 = 0.16
v. d_5 = 0.4275

2. Declare Variables
    a. EstimateTemp = 0
    b. Temperature = 0
    c. username = ""
    d. Location = ""
    e. HeatIndex = 0
    f. WindChill = 0
    g. RelativeHumidity = 0
    h. WindVelocity = 0
    i. latitude = 0.0000
    j. longitude = 0.0000
    k. Part = ""
    l. AmericanUnits = ""

3. Intro

4. Get Inputs
    a. Name
    b. Zip

5. API_ZIP
    a. Key
    b. Endpoint
    c. Retrieve
        i. Place
        ii. Longitude
        iii. Latitude

6. API
    a. Key
    b. Endpoint
    c. Retrieve
        i. Temperature
        ii. Relative Humidity
        iii. Wind Velocity

7. Calculations
    a. IF temperature > 80 report Heat Index
    b. IF temperature < 50 report Wind Chill
    c. HI = -42.379 + 2.04901523*T + 10.14333127*RH - 0.22475541*T*RH - 0.00683783*T*T - 0.05481717*RH*RH + 0.00122874*T*T*RH + 0.00085282*T*RH*RH - 0.00000199*T*T*RH*RH

d. CI = 35.74 + 0.6215*T - 35.75*(V^0.16) + 0.4275*T*(V^0.16)

8. Output
    a. Display Temperature
    b. Display Heat Index / Wind Chill

Other functions (starting with Assignment 2):

1. Print_introduction()
    a. Prints information about the program

2. Get_name()
    a. Prompt the user for name
    b. Return name

3. Get_Location(zip_code)
    a. Prompt for zip code (integer)
    b. Return zip

4. API_geocode()
    a. Use API to convert zip to LON-LAT
    b. Return LON-LAT

5. get_longitude()
    a. Store longitude from API response
    b. Return longitude

6. get_latitude()
    a. Store latitude from API response
    b. Return latitude

7. get_place()
    a. Store place from API response
    b. Return place

8. API_weather()
    a. Use API to retrieve current conditions

9. get_wind velocity
    a. Store wind velocity from API response
    b. Return wind velocity

10. Get_temperature
    a. Store temperature from API response
    b. Return temperature

11. get_relative humidity
    a. Store humidity from API response

b. Return relative humidity

12. Calc_heat_index(temp, humidity)
    a. Calculates heat index

13. Calc_wind_chill(wind velocity, temperature)
    a. Calculates wind chill

14. OutPuts()
    a. If
        i. 50 or below
        ii. 80 or above
        iii. Between 51 and 79
    b. Print goodbye message