# Test plan

## Date Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check constructor correctly initializes the data, and all getters can retrieve the data | Default Constructor  Day: -1  Month: -1  Year: -1  Constructor  Day: 2  Month: 2  Year: 2000 | Pass |
| 2 | Check that day setter are working correctly | Test Day setter Day: 24 | Pass |
| 3 | Check that month setter are working correctly | Test Month setter Month: 10 | Pass |
| 4 | Check that year setter are working correctly | Test Year setter Year: 1995 | Pass |

A screen shot of a computer

Description automatically generated

## Time Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check constructor correctly initializes the data, and all getters can retrieve the data | Default Constructor  Hour: -1  Minute: -1  Constructor  Hour: 10  Minute: 30 | Pass |
| 2 | Check that hour setter are working correctly | Test Hour setter Hour: 9 | Pass |
| 3 | Check that minute setter are working correctly | Test Minute setter Minute: 56 | Pass |
| 4 | Check that < operator overload function can correctly compare 2 Time objects | Test < operator.  Test Time 1: 05:05  Test Time 2: 09:56  Compare testTime 1 < testTime 2: 0 | Pass |
| 5 | Check that > operator overload function can correctly compare 2 Time objects | Test > operator Compare testTime 1 > testTime 2: 1 | Pass |
| 6 | Check that == operator overload function can correctly equate 2 Time objects | Test == operator  Check testTime 1 == testTime 2: 0 | Pass |

A screenshot of a computer

Description automatically generated

## Weather Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that constructor correctly initialize the data and all getters can retrieve the data | Check constructor.  Date: -1/-1/-1  Time: 0-1:0-1  Wind Speed: -1  Solar Radiation-1  Air Temperature: -1 | Pass |
| 2 | Check that Date setter is working correctly | Date: 16/3/2016 | Pass |
| 3 | Check that Time setter is working correctly | Time: 09:15 | Pass |
| 4 | Check that Wind Speed setter is working correctly | Wind Speed: 11.1 | Pass |
| 5 | Check that Solar Radiation setter is working correctly | Solar Radiation: 23.15 | Pass |
| 6 | Check that Air Temperature setter is working correctly | Air Temperature: 78.25 | Pass |
| 7 | Check that < operator can correctly compare the time of weather object | Test < operator.  Time of weather 1: 0-1:0-1  Time of weather 2: 09:15  Compare weather1 < weather2: 1 | Pass |
| 8 | Check that > operator can correctly compare the time of weather object | Test > operator.  Time of weather 1: 0-1:0-1  Time of weather 2: 09:15  Compare weather1 > weather2: 0 | Pass |
| 9 | Check that == operator can correctly equate the time of weather object | Test == operator  Time of weather 1: 0-1:0-1  Time of weather 2: 09:15  Chack weather 1 == weather 2: 0 | Pass |

A black rectangular object with white text

Description automatically generated

## BstWeatherHelper Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that GetWeatherWindSpeed function can correctly get the wind speed data from given weather object | Test GetWeatherWindSpeed  11.1 | Pass |
| 2 | Check that GetWeatherAirTemp function can correctly get the air temperature data from given weather object | Test GetWeatherAirTemp  78.25 | Pass |
| 3 | Check that GetWeatherSolarRad function can correctly get the solar radiation data | Test GetWeatherSolarRad  23.15 | Pass |

A black screen with a white border

Description automatically generated

## BST Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that Constructor can correctly initialize an empty BST and TreeNodeCount function would return 0 | Test Constructor  0 | Pass |
| 2 | Check that the copy constructor can correctly deep copy another Vector | Test Copy constructor.  Tree1 values: 1  Tree2 values after copying Tree1 and deleting Tree1: 1 | Pass |
| 3 | Check that the deconstructor can successfully delete the Vector | Test deconstructor.  Tree1 values:  Tree2 values: 1  Tree2 values after decontructed: | Pass |
| 4 | Check that BST is correctly deleted using DeleteTree function | Test DeleteTree function.  Tree1 values:  Tree1 values after deleting tree: | Pass |
| 5 | Check that Operator= can correctly deep copy another Vector | Test = operator  Tree1 values: 1  Tree2 values after copying Tree1 and deleting Tree1: 1 | Pass |
| 6 | Check that Search function can correctly determine if a value exist in BST | Test Search function.  Value found: 1.  Value not found 0 | Pass |
| 7 | Check that Insert function can correctly insert into BST | Test Insert function.  Tree1 values:  After insert  Tree1 values: 1 2 3 4 5 | Pass |
| 8 | Check that DeleteNode function can correctly delete a node in BST by value given | Test DeleteNode function.  Tree1 values: 1 2 3 4 5  Deletes node.  Tree1 values after delete: 2 3 4 5 | Pass |
| 9 | Check that TreeNodeCount function can return the correct number of nodes in BST | Test TreeNodeCount function.  Count: 4 | Pass |
| 10 | Check that InOrderTraversal function can traverse the BST in an in-order way | Test InOrderTraversal function.  1 2 3 4 5 | Pass |
| 11 | Check that PreOrderTraversal function can traverse the BST in a pre-order way | Test PreOrderTraversal function.  2 1 5 4 3 | Pass |
| 12 | Check that PostOrderTraversal function can traverse the BST in a post order way | Test PostOrderTraversal function.  2 1 5 4 3 | Pass |
| 14 | Check that GetSumValueFloat function can correctly get the total value of all nodes in BST | Test GetSumValueFloat function.  15  Test GetSumValueFloat with function as parameter.  23.1 | Pass |
| 15 | Check that InOrderTraversalFloat function can correctly get the value of individual nodes in BST | Test InOrderTraversalFloat function.  53.25 |  |

**Test 1-8**

A black screen with white text

Description automatically generated

**Test 9-15**

A black screen with white text

Description automatically generated

## Logic Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that LoadMultiDataFileCSV can correctly load multiple data file into map using an index file | Test LoadMultiDataFileCSV function.  Loading file: dataFile1.csv  Loaded data file: dataFile1.csv.  Loading file: dataFile2.csv  Loaded data file: dataFile2.csv.  1/1/2012 08:20  1/1/2012 08:30  1/1/2012 08:40  1/1/2012 08:50  1/1/2012 09:00 | Pass |
| 2 | Check that LoadDataFileCSV can correctly load a data file into map using an index file | Test LoadDataFileCSV function.  17/8/2010 19:10  17/8/2010 19:20  17/8/2010 19:30  17/8/2010 19:40  17/8/2010 19:50 | Pass |
| 3 | Check that InsertWeatherToMap can correctly insert a Weather Object into map | Test InsertWeatherToMap function.  16/3/2016 09:15 | Pass |
| 4 | Check that RemoveLeadingZeros function can correctly remove the leading zeros when given a string | Test RemoveLeadingZeros (string)  Test RemoveLeadingZeros (08): 8  Test RemoveLeadingZeros (08.9): 8.9 | Pass |
| 5 | Check that IntMonthToString function can correctly return the month in string when provided with a month in integer | Test IntMonthToString (unsigned int)  Test IntMonthToString (8): August  Test IntMonthToString (13):  Error converting int month to string.  ERROR | Pass |
| 6 | Check that ConvertWindMStoKMH function can correctly convert windspeed provided in m/s to km/h | Test ConvertWindMStoKMH (float)  Test ConvertWindMStoKMH (10): 36  Test ConvertWindMStoKMH (10.5): 37.8 | Pass |
| 7 | Check that ConvertSolarRadiationWMtoKWH function can correctly convert solar radiation provided in W/m2 to kWh/m2 | Test ConvertSolarRadiationWMtoKWH (float)  Test ConvertSolarRadiationWMtoKWH (120): 0.02  Test ConvertSolarRadiationWMtoKWH (120.42): 0.02007 | Pass |
| 8 | Check that AverageWindSpeedMonth function can correctly find and calculate the average windspeed for a month | Test AverageWindSpeedMonth (vector, int, int)  Test AverageWindSpeedMonth (TestLog,3,2016): 0.25.  Test AverageWindSpeedMonth (TestLog,1,2010): 6.25 | Pass |
| 9 | Check that StandardDeviationWindSpeed function can correctly find and calculate the standard deviation of wind speed for a month | Test StandardDeviationWindSpeed (vector, int, int)  Test StandardDeviationWindSpeed (TestLog,6,2011): 0.433013.  Test StandardDeviationWindSpeed (TestLog,1,2010): 0.433013 | Pass |
| 10 | Check that AverageAirTemperatureMonth function can correctly find and calculate the average ambient air temperature for a month | Test AverageAirTemperatureMonth (vector, int, int)  Test AverageAirTemperatureMonth (TestLog,6,2011): 8.5475.  Test AverageAirTemperatureMonth (TestLog,1,2010): 21.71 | Pass |
| 11 | Check that StandardDeviationAirTemperature function can correctly find and calculate the standard deviation of ambient air temperature for a month | Test StandardDeviationAirTemperature (vector, int, int)  Test StandardDeviationAirTemperatue (TestLog,6,2011): 0.157063.  Test StandardDeviationAirTemperatue (TestLog,1,2010): 0.58809 | Pass |
| 12 | Check that SumSolarRadiationMonth function can correctly find and calculate the total solar radiation for a month | Test SumSolarRadiationMonth (vector, int, int)  Test SumSolarRadiationMonth (TestLog,6,2011): 0.  Test SumSolarRadiationMonth (TestLog,1,2010): 3470 | Pass |
| 13 | Check that WriteAppendFile able to correctly write to "WindTempSolar.csv" | Test WriteAppendFile (string)  \*Look for the test.csv file  teststring  teststring2 | Pass |
| 14 | Check that calculateSPCC able to correct calculate the Sample Pearson Correlation Coefficient for x and y float vectors. | Test calculateSPCC (vector, vector)  SPCC: 1 | Pass |
| 15 | Check that extractMonthData able to extract wind speed, temperature and solar radiation of a month and store them in vectors. | Test extractMonthData with Edge Cases  Month and Year Not Found Test: Wind Speeds Size: 4, Air Temperatures Size: 4, Solar Radiations Size: 4 | Pass |

**Test 1-7**

**A black background with a white border

Description automatically generated**

**Test 8-15**

A computer screen with white text

Description automatically generated

**Test 13**

**A screenshot of a computer

Description automatically generated**

## Validator Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that IsValidDate function can correctly determine if a date is valid when given the day, month, and year | Checking IsValidDate (int day, int month, int year)  Positive case (29/2/2020): 1  Negative case, wrong day (31/2/2020): 0  Negative case, wrong month (29/13/2020): 0  Negative case, wrong year (29/2/99): 0  Negative case, not leap year (29/2/2022): 0 | Pass |
| 2 | Check that IsValidDate function can correctly determine if a date is valid when given the month and year only | Checking IsValidDate (int month, int year)  Positive case (2/2020): 1  Negative case, wrong month (13/2020): 0  Negative case, wrong year (2/99): 0 | Pass |
| 3 | Check that IsValidMonth function can correctly determine if a int is valid month | Checking IsValidMonth (int month)  Positive case (2): 1  Negative case, wrong month (13): 0 | Pass |
| 4 | Check that IsLeapYear function can correctly determine if a year is a leap year | Checking IsLeapYear (int year)  Positive case (2020): 1  Negative case (2022): 0 | Pass |
| 5 | Check that IsValid24HourTime function can correctly determine if a given hour and minute is valid 24-hour time | Checking IsValid24HourTime (int hour, int minute)  Positive case (8,50): 1  Negative case, wrong hour (25,50): 0  Negative case, wrong minute (24,60): 0 | Pass |
| 6 | Check that IsNumber function can correctly determine if a provided string is positive int/float | Checking IsNumber (string input)  Positive case (415): 1  Positive case (63.41): 1  Negative case, negative int (-1): 0  Negative case, negative float (-1.2): 0  Negative case, too many decimal points (123456.123456789012345): 1  Negative case, octal numbers (08): 0 | Pass |
| 7 | Check that IsInteger function can correctly determine if a provided string is a positive integer | Checking IsInteger (string input)  Positive case (415): 1  Negative case (45.21): 0  Negative case (-1): 0  Negative case (08): 0 | Pass |

A screenshot of a computer

Description automatically generated

## FunctionHelper Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that DisplayMenu function can successfully display the menu | Test DisplayMenu function.  1. Average wind speed in km/h and standard deviation for a given specified month and year.  2. Average ambient air temperature and sample standard deviation for each month of a specified year.  3. sPCC for a given specific month for all years that have been loaded.  4. Write Average wind speed in km/h (standard deviation), average ambient air temperature (standard deviation) and total solar radiation in KWH/M2 for each month of a specified year to CSV.  5. Reload data from data/data\_source.txt file.  6. Exit the program. | Pass |
| 2 | Check that FindAverageWindTempMonth can correctly carry out the function of option 1 of menu given map, month, and year input | Test FindAverageAirTemperatureAndStandardDeviationMonth function for option 1.  January 2010: Average:21.709999 degrees C, stdev:0.588090 | Pass |
| 3 | Check that FindAverageWindTempMonth can correctly carry out the function of option 2 of menu given map and year input | Test FindAverageTemperatureYear function for option 2.  January 2010: Average:21.709999degrees C, stdev:0.588090  February 2010: No Data  March 2010: Average:18.940001degrees C, stdev:0.052915  April 2010: No Data  May 2010: No Data  June 2010: No Data  July 2010: No Data  August 2010: Average:11.366000degrees C, stdev:0.226504  September 2010: No Data  October 2010: No Data  November 2010: No Data  December 2010: No Data  October 2010: No Data  November 2010: No Data  December 2010: No Data | Pass |
| 4 | Check that FindsPCCForGivenMonth can correctly carry out the function of option 3 of menu given map and month input of all year loaded | Test FindsPCCForGivenMonth function for option 3.  Average Sample Pearson Correlation Coefficient for month January  S\_T: 0.303070  S\_R: -0.185711  T\_R: 0.468130 | Pass |
| 5 | Check that OutputFileAverageWindTempSolarMonth can correctly carry out the function of option 4 of menu given map and year input | 2010  January,22.500000(0.433013),21.709999(0.588090),0.578333  March,21.600000(0.000000),18.940001(0.052915),0.000000  August,7.200000(1.095445),11.366000(0.226504),0.000000 | Pass |
| 6 | Check that Option 5 can correctly carry out the function of option 4 of menu given map and year input | Test OutputFileAverageWindTempSolarMonth function for option 4.  Loading file: test2.csv  Loaded data file: test2.csv | Pass |

A screenshot of a computer program

Description automatically generated

A computer screen with numbers and letters

Description automatically generated

## Vector Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Test default constructor | Default Constructor  Size: 0  Capacity: 0  Is Empty: Yes | Pass |
| 2 | Test constructor with size | Constructor with size  Size: 5  Capacity: 5  Is Empty: No | Pass |
| 3 | Test constructor with size and initial value | Constructor with size and initial value  Size: 5  Capacity: 5  Is Empty: No | Pass |
| 4 | Test element access | Element Access  First Element: 20  Second Element: 10 | Pass |
| 5 | Test push\_back | Push Back  Size: 6  Capacity: 10  Last Element: 30 | Pass |
| 6 | Test pop\_back | Pop Back  Size: 5  Capacity: 10 | Pass |
| 7 | Test reserve | Reserve  Size: 5  Capacity: 10 | Pass |
| 8 | Test resize | Resize  Size: 3  Capacity: 10 | Pass |
| 9 | Test iterators | Iterators  20 10 10 | Pass |

A black screen with white text

Description automatically generated

## Main SIT (Using Test.csv)

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that if user entered invalid file name in data/data\_source.txt file, the program would print error and continue with available data | Error while opening file: notfound.csv | Pass |
| 2 | Check that invalid month or year input while using function 1, it will result in error | Error: invalid month. Expects integer. | Pass |
| 3 | Check that invalid month and year combination will result in error and stopping of program | Error: invalid date. | Pass |
| 4 | Check that function 1 can print correctly average wind speed and standard deviation when data is found for the specific month and year in km/h respectively | January 2016: Average Speed:28.313869 km/h stdev:2.693603 | Pass |
| 5 | Check that function 1 will show no data if no data is found for the specific month and year | February 2010: No Data | Pass |
| 6 | Check that if user entered invalid year while using function 2, it will result in error | Error: invalid year. Expects integer. | Pass |
| 7 | Check that function 2 can correctly print the air temperature and standard deviation of all 12 months in the specific year in km/h and degrees C respectively and no data if no data is found for the month | January 2016: Average:27.320732degrees C, stdev:3.911447  February 2016: No Data  March 2016: No Data  April 2016: No Data  May 2016: No Data  June 2016: No Data  July 2016: No Data  August 2016: No Data  September 2016: No Data  October 2016: No Data  November 2016: No Data  December 2016: No Data | Pass |
| 8 | Check that if user entered invalid year while using function 3, it will result in error | Error: invalid month. Expects integer. | Pass |
| 9 | Check that function 3 can correctly output the SPCC calculation of 3 different combinations of user specified month of all year loaded. | Average Sample Pearson Correlation Coefficient for month January  S\_T: 0.098138  S\_R: 0.298749  T\_R: 0.624492 | Pass |
| 10 | Check that if user entered invalid year while using function 4, it will result in error | Error: invalid year. Expects integer. | Pass |
| 11 | Check that function 4 can correctly output the average wind speed, standard deviation of wind speed, average air temperature, standard deviation of air temperature and total solar radiation for each month of a specific year in km/h, degrees C and kWh/m2 and no line for months with no data. | 2016 january,28.313869(2.693603),  27.320732(3.911447),15.915333 | Pass |
| 12 | Check that function 5 can clear the memory and reload the menu with different files that user put in data\_source.txt file. | Loading file: test1.csv  Loaded data file: test1.csv.  2016  January 2016: Average Speed: 28.313869 km/h  stdev: 2.693603  Average Sample Pearson Correlation Coefficient for month January  S\_T: 0.098138  S\_R: 0.298749  T\_R: 0.624492  Loading file: test2.csv  Loaded data file: test2.csv.  January 2009: Average Speed: 22.270073 km/h  stdev: 2.407035  Average Sample Pearson Correlation Coefficient for month January  S\_T: 0.481042  S\_R: 0.410883  T\_R: 0.872047 | Pass |
| 13 | Check that function 6 can quit the program | \*Program exits | Pass |
| 14 | Check that inputting any function other than 1-6 will return in error and repeatedly display the menu | Error: Unknown command. Only numbers 1-6 accepted. |  |

**Test 1**

A screenshot of a computer program

Description automatically generated

**Test 2-3**

A screenshot of a computer

Description automatically generated

**Test 4**

**A screenshot of a computer program

Description automatically generated**

**Test 5-6**

**A screenshot of a computer program

Description automatically generated**

**Test 7-8**

**A screen shot of a computer

Description automatically generated**

**Test 9**

**A screenshot of a computer error

Description automatically generated**

**Test 10-11**

**A black screen with white text

Description automatically generated**

**A computer screen with numbers

Description automatically generated**

**Test 12**

**A computer screen with text on it

Description automatically generated**

**A black screen with white text

Description automatically generated**

**Test 13-14**

**A black screen with text on it

Description automatically generated**

## More Test for Main Sit (Unit Testing)

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that it can read multiple file name together with empty line.  MetData-31-3b.csv  MetData-31-3c.csv | Load without no problem. | Pass |
| 2 | Check if it can read line with no value and only having commas.  ,,,,,,,,,,,, | Warning: Line of commas detected at row 71  Warning: Line of commas detected at row 71 | Pass |
| 3 | Check if it can read multiple files together some error fine.  Metdata-Jan-Dec2007.csv  Metdata-Jan-Dec2008.csv  Metdata-Jan-Dec2010.csv | Loading file: Metdata-Jan-Dec2007.csv  Loading file: Metdata-Jan-Dec2008.csv  Error while opening file: Metdata-Jan-Dec2010.csv | Pass |
| 4 | Check if it can read multiple files and all the function work properly.  MetData\_Mar01-2015-Mar01-2016-ALL.csv  MetData\_Jan01-2010-Jan01-2011-ALL.csv  MetData-31-3b.csv  MetData\_Jan01-2011-Jan01-2012-ALL.csv  MetData-31-3c.csv  MetData\_Mar01-2014-Mar01-2015-ALL.csv | Loading file: MetData\_Mar01-2015-Mar01-2016-ALL.csv  Loading file: MetData\_Jan01-2010-Jan01-2011-ALL.csv  Loading file: MetData-31-3b.csv  Loading file: MetData\_Jan01-2011-Jan01-2012-ALL.csv  Loading file: MetData-31-3c.csv  Loading file: MetData\_Mar01-2014-Mar01-2015-ALL.csv | Pass |

**Test 1-3**

**A black background with white text

Description automatically generated**

**A computer screen shot of text

Description automatically generated**

**Test 4**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**