ELECTRONIC ASSIGNMENT COVERSHEET

Murdoch UNIVERSITY

Kaplan Singapore

Student Number	Murdoch: 34408441 Kaplan: CT0362378
Surname	Lim
Given name	Seng Wei
Email	seng.wei.lim@outlook.com

Unit Code	ICT 283
Unit name	Data Structures and Abstractions
Enrolment mode	PT
Date	3 Mar 2022
Assignment number	1
Assignment name	Assignment 1
Tutor	Kang Leng

Student's Declaration:

- Except where indicated, the work I am submitting in this assignment is my own work and has not been submitted for assessment in another unit.
- This submission complies with Murdoch University's academic integrity commitments. I am aware that information about plagiarism and associated penalties can be found at http://www.murdoch.edu.au/teach/plagiarism/. If I have any doubts or queries about this, I am further aware that I can contact my Unit Coordinator prior to submitting the assignment.
- I acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:
  -reproduce this assignment and provide a copy to another academic staff member; and/or
  -submit a copy of this assignment to a plagiarism-checking service. This web-based service may
  -
  -retain a copy of this work for the sole purpose of subsequent plagiarism checking, but has a legal agreement with the University that it will not share or reproduce it in any form.
- I have retained a copy of this assignment.

I am aware that I am making this declaration by submitting this document electronically and by using my Murdoch ID and password it is deemed equivalent to executing this declaration with my written signature.

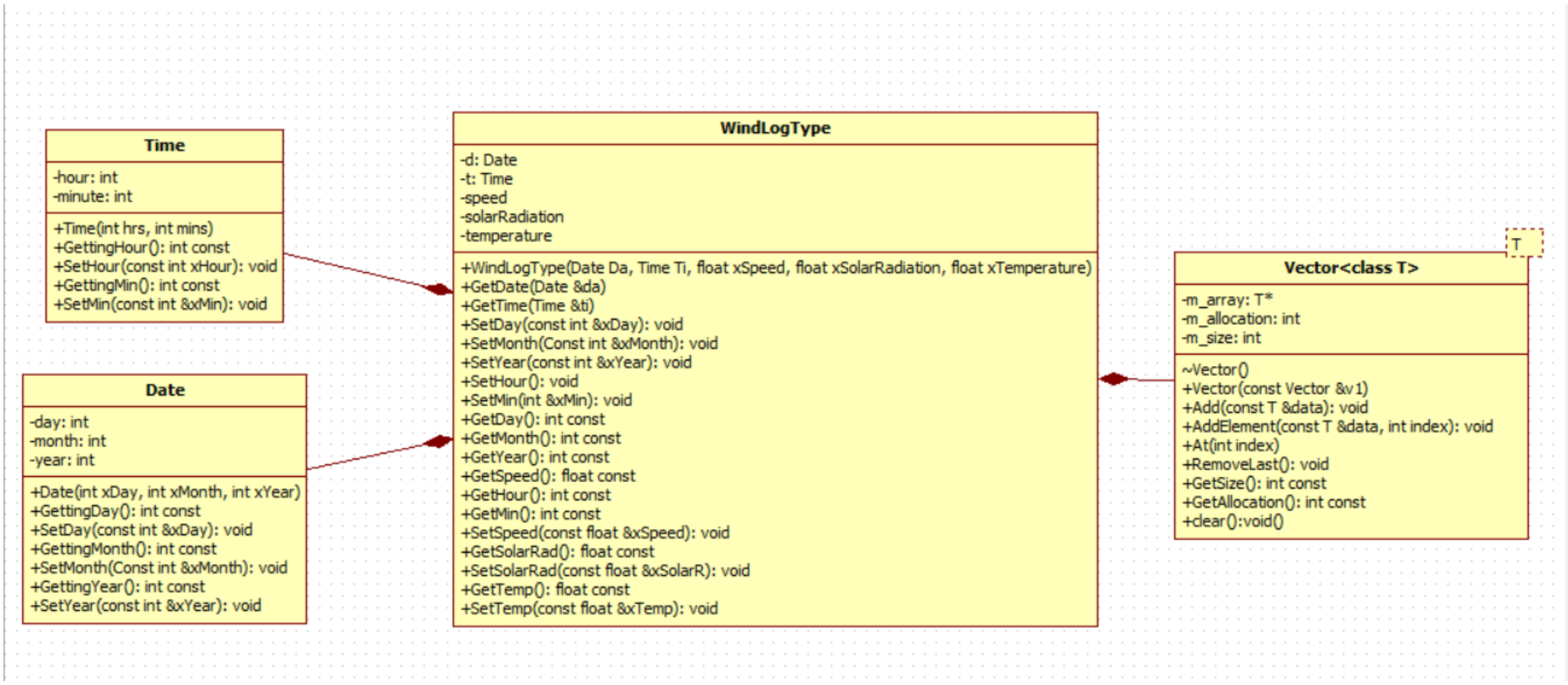
Signed LIM SENG WEI

(Write your name in the space above)

Title: ICT283 Assignment 1

Name: Lim Seng Wei
Class: ICT 283 PT B

1. UML Class Diagram


Time
-hour: int
-minute: int
+Time(int hrs, int mins)
+GettingHour(): int const
+SetHour(const int xHour): void
+GettingMin(): int const
+SetMin(const int &xMin): void
Date
-day: int
-month: int
-year: int
+Date(int xDay, int xMonth, int xYear)
+GettingDay(): int const
+SetDay(const int &xDay): void
+GettingMonth(): int const
+SetMonth(Const int &xMonth): void
+GettingYear(): int const
+SetYear(const int &xYear): void
WindLogType
-d: Date
-t: Time
-speed
-solarRadiation
-temperature
+WindLogType(Date Da, Time Ti, float xSpeed, float xSolarRadiation, float xTemperature)
+GetDate(Date &da)
+GetTime(Time &ti)
+SetDay(const int &xDay): void
+SetMonth(Const int &xMonth): void
+SetYear(const int &xYear): void
+SetHour(): void
+SetMin(int &xMin): void
+GetDay(): int const
+GetMonth(): int const
+GetYear(): int const
+GetSpeed(): float const
+GetHour(): int const
+GetMin(): int const
+SetSpeed(const float &xSpeed): void
+GetSolarRad(): float const
+SetSolarRad(const float &xSolarR): void
+GetTemp(): float const
+SetTemp(const float &xTemp): void
T
Vector<class T>
-m_array: T*
-m_allocation: int
-m_size: int
~Vector()
+Vector(const Vector &v1)
+Add(const T &data): void
+AddElement(const T &data, int index): void
+At(int index)
+RemoveLast(): void
+GetSize(): int const
+GetAllocation(): int const
+clear():void()

2. Data Dictionary

Design rationale: Have changed windlog struct to a windlog class instead for better encapsulation and future proofing. Getters and setters can all be accessed from main and client has no access to Date and Time class. Have converted each option choice to be part of a function, reading csv into the vector is also converted into a function for a short and simple main().

Date Class

Name	Type	Protection	Description	Rationale
Date			Simulates a date	
day	integer	-	Value of day from a date	
month	integer	-	Value of month from a date	
year	integer	-	Value of year from a date	
Date()	integer	+	Default constructor	
Date(int xDay,int xMonth, int xYear)	integer	+	Constructor to set values	
GettingDay()	integer	+	Getter method	
SetDay(int &xDay)	void	+	Setter method	
GettingMonth()	integer	+	Getter method	
SetMonth(int &xMonth)	void	+	Setter method	
GettingYear()	integer	+	Getter method	
SetYear(int &xYear)	void	+	Setter method	

Time Class

Name	Type	Protection	Description	Rationale
Time			Simulates the time	
hour	integer	-	Value of hour from a time	
minute	integer	-	Value of minutes from a time	
Time()	integer	+	Default constructor	
Time(int xHour,int xMin)	integer	+	Constructor to set values	
GettingHour()	integer	+	Getter method	
SetHour(int &xHour)	void	+	Setter method	
GettingMin()	integer	+	Getter method	
SetMin(int &xMin)	void	+	Setter method	

WindLogType Class

Name	Type	Protection	Description	Rationale
WindLogType			Simulates a set of data from the data file	
d	Date object	-	Date object	Date class methods are now private, will create getter/setter for its method in this class
t	Time object	-	Time object	Time class methods are now private, will create getter/setter for its method in this class
speed	float	-	Value of speed from the data file	
solarRadiation	float	-	Value of solar radiation from the data file	
temperature	float	-	Value of temperature from the data file	
WindLogType()	void	+	Default constructor	
WindLogType (Date Da, Time Ti, float xSpeed, float xSolarRadiation, float xTemperature)	void	+	Constructor to set values	
GetDate(Date &Da)	void	+	Getter method	Get the date object, used in output stream overloading in main.cpp
GetTime(Time &Ti)	void	+	Getter method	Get the time object, used in output stream overloading in main.cpp
SetDay(int &xDay)	void	+	Setter method	Set the value of day into date object
SetMonth(int &xMonth)	void	+	Setter method	Set the value of month into date object
SetYear(int &xYear)	void	+	Setter method	Set the value of year into date object
SetHour(int &xHour)	void	+	Setter method	Set the value of hour into time object
SetMin(int &xMin)	void	+	Setter method	Set the value of minute into time object
GetDay()	integer	+	Getter method	Return day from date object, used when looping through vector to find specific date
GetMonth()	integer	+	Getter method	Return month from date object, used when looping through vector to find specific date
GetYear()	integer	+	Getter method	Return year from date object, used when looping

				through vector to find specific date
GetSpeed ()	float	+	Getter method	
SetSpeed(float &xSpeed)	float	+	Setter method	
GetSolarRad ()	float	+	Getter method	
SetSolarRad(float &xSolarR)	float	+	Setter method	
GetTemp ()	integer	+	Getter method	
SetTemp(float &xTemp)	float	+	Setter method	

Vector Template Class

Name	Type	Protection	Description	Rationale
T* m_array	pointer	-	Pointer to vector	Stores the address of vector
m_allocation	integer	-	Storage of the vector	Stores the maximum allocation of the vector
m_size	integer	-	Number of elements present in the vector	Stores the current number of elements used in the vector
Vector()	class	+	Default constructor	
~Vector()	class	+	Destructor	
Vector(const Vector &v1)	class	+	Copy constructor	Deep copying
Add()	void	+	Add an element to the next available index	Simulate push function of a vector
At (int index)	class	+	Extract element at a specific index of the vector	Allows access to every single element easily and for using in a loop
RemoveLast()	void	+	Remove last element from vector	
GetSize()	integer	+	Get the size of the vector, return m_size	To be used to loop through vector elements that are not empty
GetAllocation()	integer	+	Get the allocation/storage of the vector, return m_allocation	
clear()	void	+	Set vector to size 0	

4. Algorithm

This program reads data from a CSV and is able to perform a few functions with the data acquired

```
Main {

Create an array of string to store all 12 months of a year, size 13
Create an array of double to store speed,temperature,solar radiation individually, where the month of the date will be the index of the array, size 13
Create a vector of <WindLogType> windlog
Create a vector of <string> column
Create object of windlog

Open file for reading
Store the first line of header information into <string> vector
Store each remaining line into WindLogType vector using setters and delimiters as comma
```

```
Use for loops to skip unneeded columns and don’t store into the vector
Add each windlog object into the vector

While loop{
Display Menu with 5 options

Option 1:
Ask user for input

For loop through the vector{
If(userinput == year && userinput == month)
Add all the instances of speed together
Add all the instance of temperature together
Add 1 to the counter for each loop

Convert total speed to be in km/h
Calculate average speed, total speed divide by counter
Calculate average temperature

if(average speed and average temperature !=0)
Output average speed and average temperature
}


Option 2:
Ask user for input

For loop through the vector{
if(userinput == year)

Extract every data that matches the year and save them into the array using the month as the index
Calculate total and average speed
Calculate total and average temperature
Convert speed to km/h
Add 1 to the counter for each loop
}

For loop 12 times starting from index 1{
if(average speed and average temperature !=0)
output the month using month array
output average speed and average temperature using their array
}


Option 3:
Ask user for input

For loop through the vector{
if(userinput == year)
if(solarRadiation >=100)
calculate total solar radiation
```

```
    convert solar radiation to be kWh/m2
}
For loop 12 times starting from index 1{
    if(total solar radiation !=0)
        output month and the total solar radiation
    if(total solar radiation ==0)
        output “No Data”
}
```

Option 4:

```
Boolean fileWritten = false;

Ask user for input

Open file for writing
For loop through the vector{
    if(userinput == year)
        Extract every data that matches the year and save them into the array using the month as the index
        Calculate total and average speed
        Calculate total and average temperature
        Convert speed to km/h
        Add 1 to the counter for each loop
}

if(solarRadiation >=100)
    calculate total solar radiation
    convert solar radiation to be kWh/m2

For loop 12 times starting from index 1{
    if(total solar radiation !=0, total speed !=0, total temperature !=0)
        write to CSV
        fileWritten = true;
}

if(fileWritten == false)
    output “No Data”
```

Option 5:

```
exit the program
```

5. Testing

Date Class

Test ID	Description	Input	Expected output	Pass/Fail
1	Test default constructor initialize correctly	Date d1;	0/0/0	Pass
2	Test Getter for day	d2.GettingDay()	3	Pass
3	Test Getter for month	d2.GettingMonth()	3	Pass
4	Test Getter for year	d2.GettingYear()	2033	Pass
5	Test Setter for day	d3.SetDay(22)	22	Pass
6	Test Setter for month	d3.SetMonth(2)	2	Pass
7	Test Setter for year	d3.SetYear(2022)	2022	Pass

Time Class

Test ID	Description	Input	Expected output	Pass/Fail
1	Test default constructor initialize correctly	Time t1;	0:00	Pass
2	Test Getter for hour	t2.GettingHour()	23	Pass
3	Test Getter for minute	t2.GettingMin()	59	Pass
4	Test Setter for hour	t3.SetHour(1)	1	Pass
5	Test Setter for minute	t3.SetMin(59)	59	Pass

WindLogType Class

Test ID	Description	Input	Expected output	Pass/Fail
1	Test default constructor initialize correctly	WindLogType w1	0 for all 5 attribute	Pass
2	Test Getter for day	w1.GetDay()	2	Pass
3	Test Getter for month	w1.GetMonth()	2	Pass
4	Test Getter for year	w1.GetYear()	2022	Pass

5	Test Getter for hour	w1.GetHour()	23	Pass
6	Test Getter for minute	w1.GetMin()	59	Pass
7	Test Getter for speed	w1.GetSpeed()	10	Pass
8	Test Getter for solar radiation	w1.GetSolarRad()	20	Pass
9	Test Getter for temperature	w1.GetTemp()	30	Pass
10	Test Setter for day	w1.SetDay(11)	11	Pass
11	Test Setter for month	w1.SetMonth(1)	1	Pass
12	Test Setter for year	w1.SetYear(2011)	2011	Pass
13	Test Setter for hour	w1.SetHour(14)	14	Pass
14	Test Setter for minute	w1.SetMin(30)	30	Pass
15	Test Setter for speed	w1.SetSpeed(20)	20	Pass
16	Test Setter for solar radiation	w1.SetSolarRad(30)	30	Pass
17	Test Setter for temperature	w1.SetTemp(40)	40	Pass

Vector Template Class

Test ID	**Description**	**Input**	**Expected output**	**Pass/Fail**
1	Test default constructor initialize correctly	Vector<int>test1	blank	Pass
2	Test Add	test1.Add(100)	100	Pass
3	Test Add element	test1.AddElement(200,0)	200	Pass

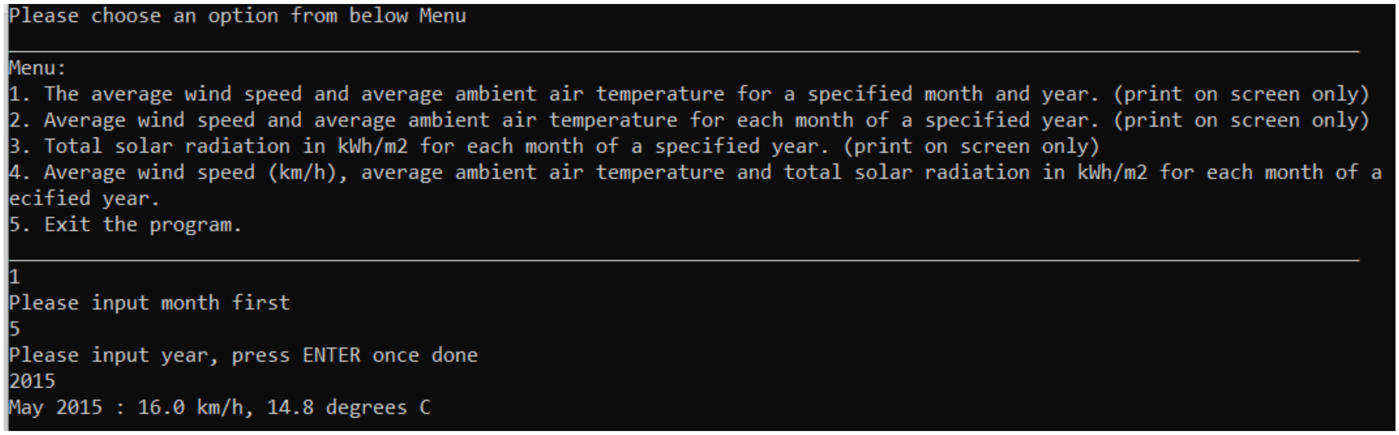
4	Test At function	test1.At(0)
		test1.At(1)
		test1.At(2)
			1
			10
			20
				Pass

5	Test remove last	test1.RemoveLast()	Vector size reduce from 3 to 2	Pass
6	Test Get size	test1.GetSize()	Return current size of 2	Pass
7	Test Get allocation	test1.GetAllocation()	Return current allocation of 5	Pass
8	Test clear	test1.clear()	Size reduced to 0, GetSize() returns 0	Pass

Main Class, tested using MetData_Mar01-2015-Mar01-2016-ALL.csv

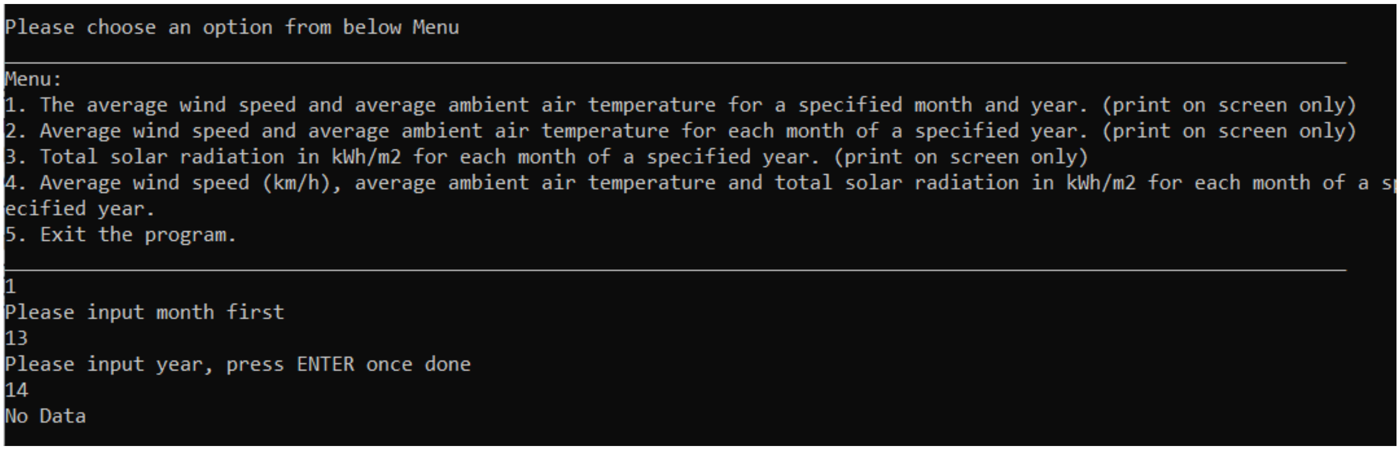
Test ID	Description	Input	Expected output	Pass/Fail
1	Test option 1 with valid input	5 2015	Shows average windspeed and temperature	Pass
2	Test option 1 with invalid input	13 14	Output no data message	Pass
3	Test option 2 with valid input	2015	Show monthly windspeed and temperature for the whole year	Pass
4	Test option 2 with invalid input	1993	Output no data message for each month	Pass
5	Test option 3 with valid input	2015	Show monthly solar radiation for the whole year	Pass
6	Test option 3 with invalid input	1993	Output no data message for each month	Pass
7	Test option 4 with valid input	2015	Output message file written to CSV	Pass
8	Test option 4 with invalid input	1993	Output no data, file written to CSV	Pass
9	Test option 5	5	Exit program	Pass
10	Enter invalid option at menu	6	Output error message	Pass

Test ID 1:



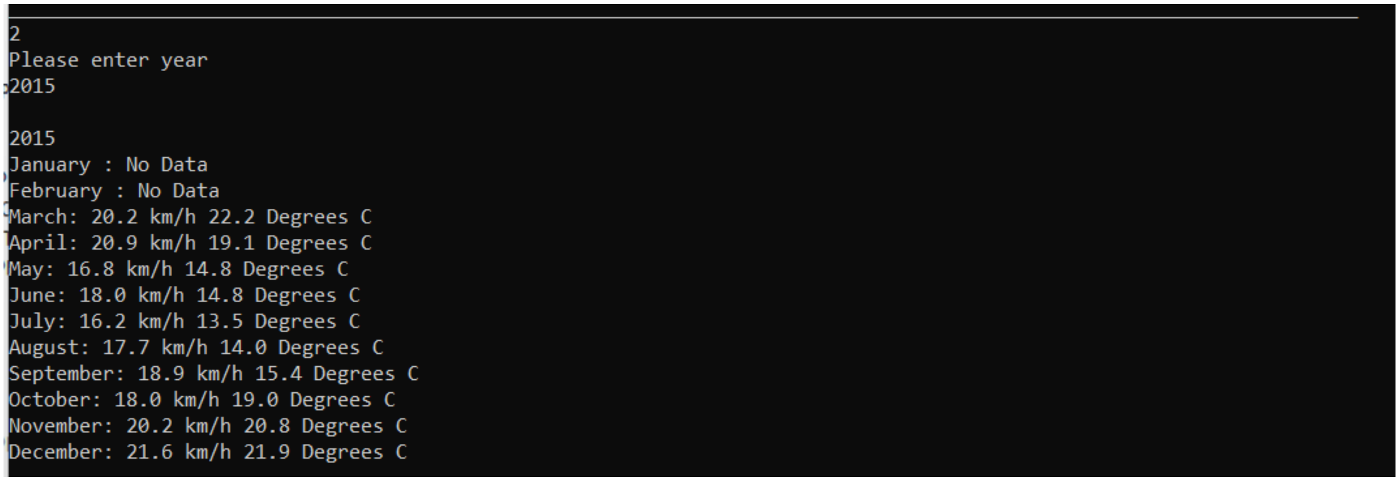
```
Please choose an option from below Menu
________________________________________________________________________________
Menu:
1. The average wind speed and average ambient air temperature for a specified month and year. (print on screen only)
2. Average wind speed and average ambient air temperature for each month of a specified year. (print on screen only)
3. Total solar radiation in kWh/m2 for each month of a specified year. (print on screen only)
4. Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month of a
ecified year.
5. Exit the program.
________________________________________________________________________________
1
Please input month first
5
Please input year, press ENTER once done
2015
May 2015 : 16.0 km/h, 14.8 degrees C
```

Test ID 2:



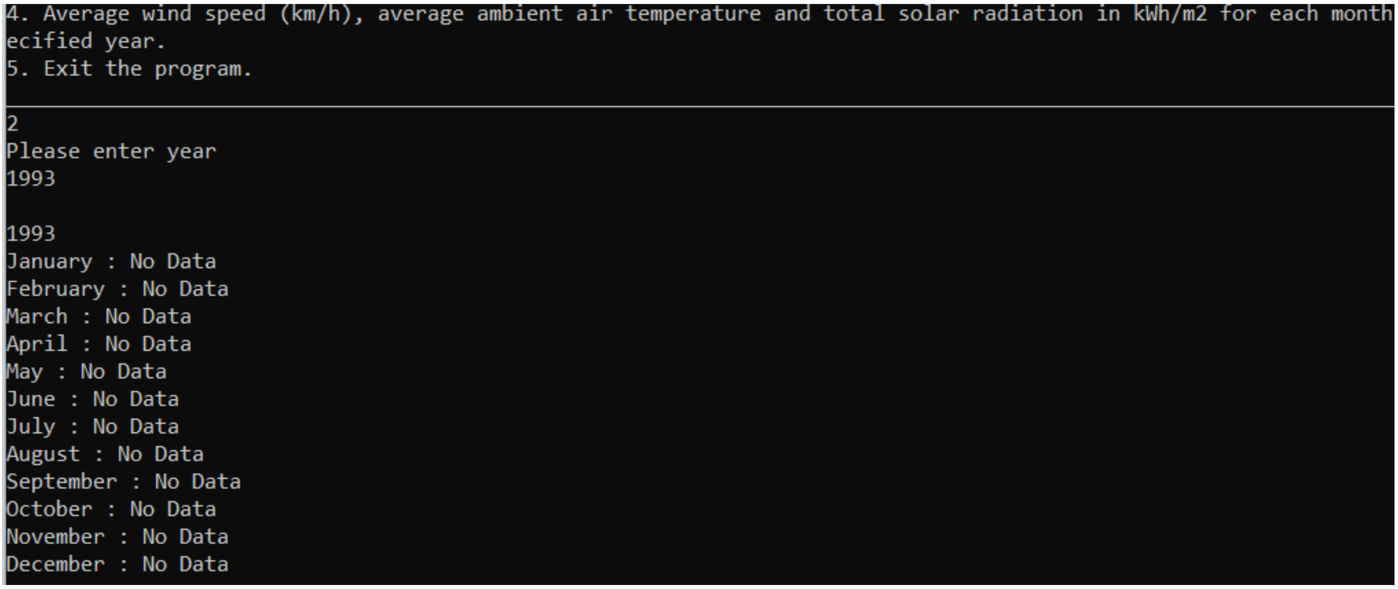
```
Please choose an option from below Menu
________________________________________________________________________________
Menu:
1. The average wind speed and average ambient air temperature for a specified month and year. (print on screen only)
2. Average wind speed and average ambient air temperature for each month of a specified year. (print on screen only)
3. Total solar radiation in kWh/m2 for each month of a specified year. (print on screen only)
4. Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month of a sp
ecified year.
5. Exit the program.
________________________________________________________________________________
1
Please input month first
13
Please input year, press ENTER once done
14
No Data
```

Test ID 3:



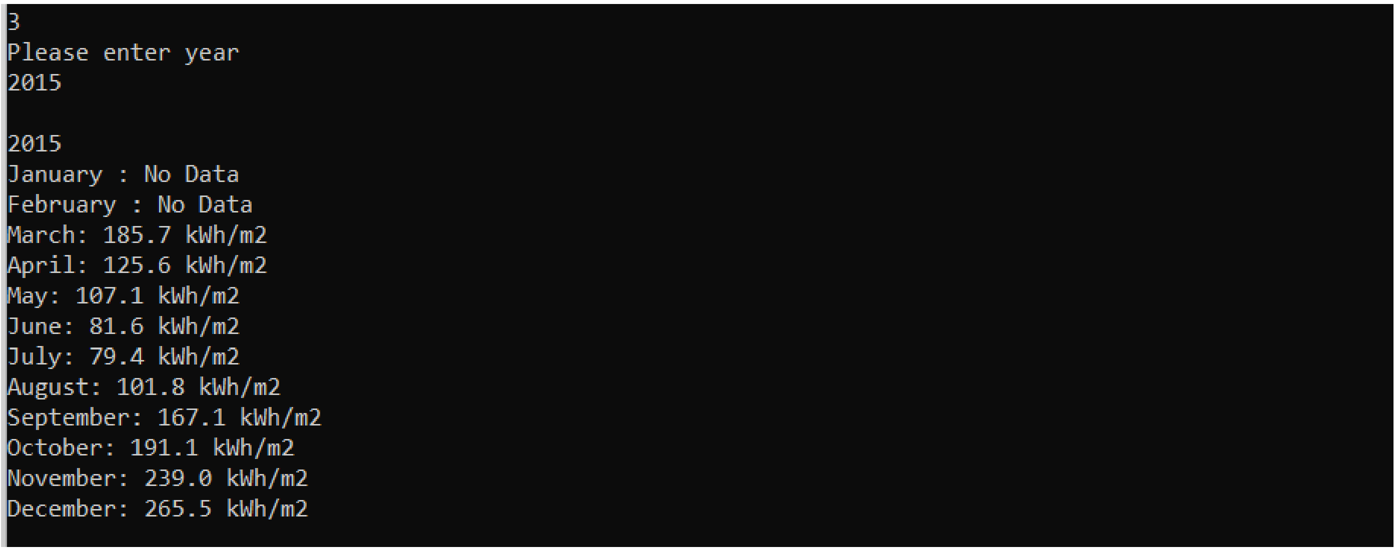
```
________________________________________________________________________________
2
Please enter year
2015

2015
January : No Data
February : No Data
March: 20.2 km/h 22.2 Degrees C
April: 20.9 km/h 19.1 Degrees C
May: 16.8 km/h 14.8 Degrees C
June: 18.0 km/h 14.8 Degrees C
July: 16.2 km/h 13.5 Degrees C
August: 17.7 km/h 14.0 Degrees C
September: 18.9 km/h 15.4 Degrees C
October: 18.0 km/h 19.0 Degrees C
November: 20.2 km/h 20.8 Degrees C
December: 21.6 km/h 21.9 Degrees C
```

Test ID 4:



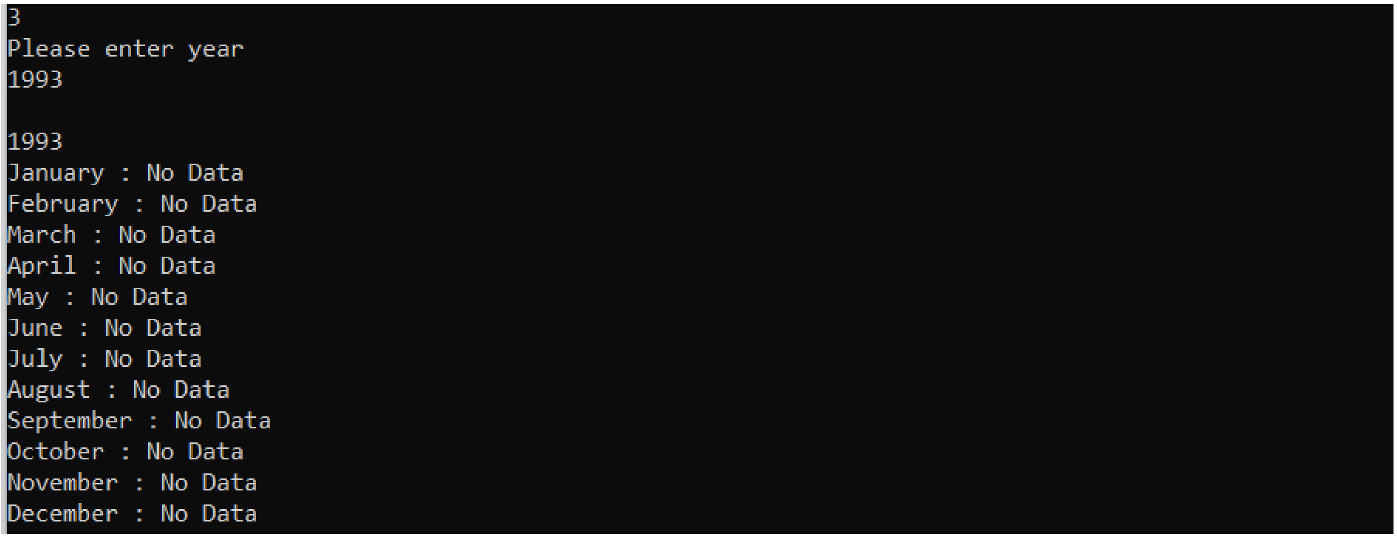
```
4. Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month
ecified year.
5. Exit the program.
________________________________________________________________________________
2
Please enter year
1993

1993
January : No Data
February : No Data
March : No Data
April : No Data
May : No Data
June : No Data
July : No Data
August : No Data
September : No Data
October : No Data
November : No Data
December : No Data
```

Test ID 5:



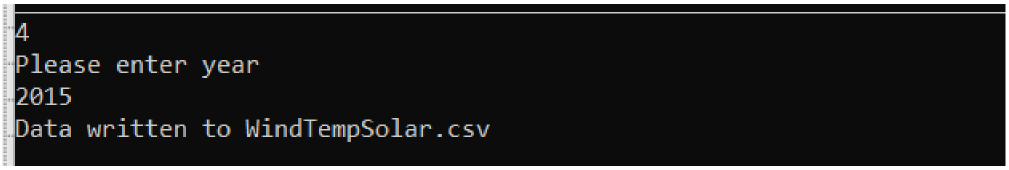
```
3
Please enter year
2015

2015
January : No Data
February : No Data
March: 185.7 kWh/m2
April: 125.6 kWh/m2
May: 107.1 kWh/m2
June: 81.6 kWh/m2
July: 79.4 kWh/m2
August: 101.8 kWh/m2
September: 167.1 kWh/m2
October: 191.1 kWh/m2
November: 239.0 kWh/m2
December: 265.5 kWh/m2
```

Test ID 6:



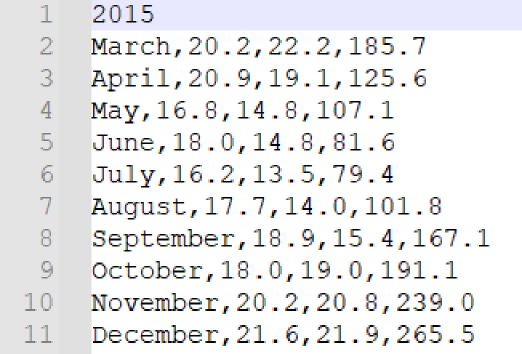
```
3
Please enter year
1993

1993
January : No Data
February : No Data
March : No Data
April : No Data
May : No Data
June : No Data
July : No Data
August : No Data
September : No Data
October : No Data
November : No Data
December : No Data
```

Test ID 7:



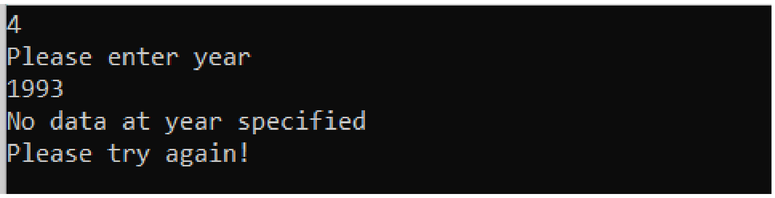
```
4
Please enter year
2015
Data written to WindTempSolar.csv
```

CSV:



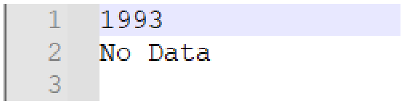
```
2015
March,20.2,22.2,185.7
April,20.9,19.1,125.6
May,16.8,14.8,107.1
June,18.0,14.8,81.6
July,16.2,13.5,79.4
August,17.7,14.0,101.8
September,18.9,15.4,167.1
October,18.0,19.0,191.1
November,20.2,20.8,239.0
December,21.6,21.9,265.5
```

Test ID 8:



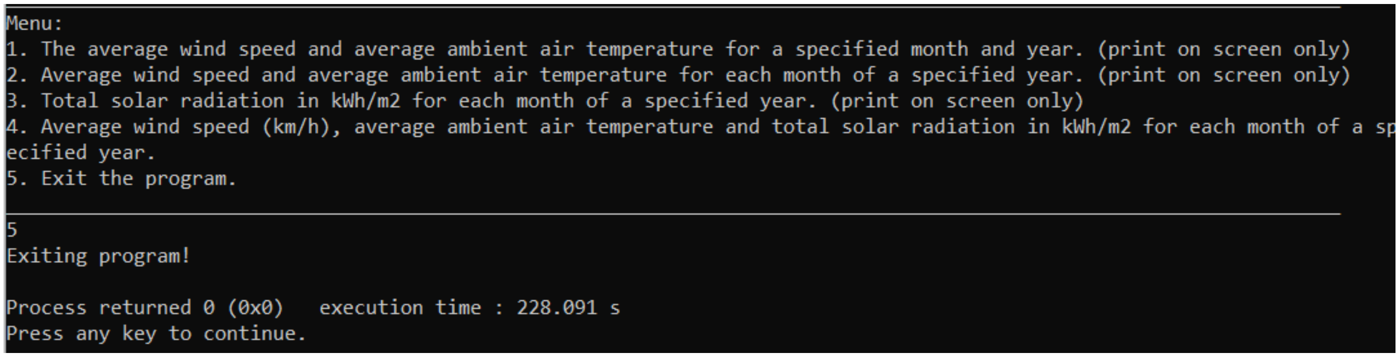
```
4
Please enter year
1993
No data at year specified
Please try again!
```

CSV:



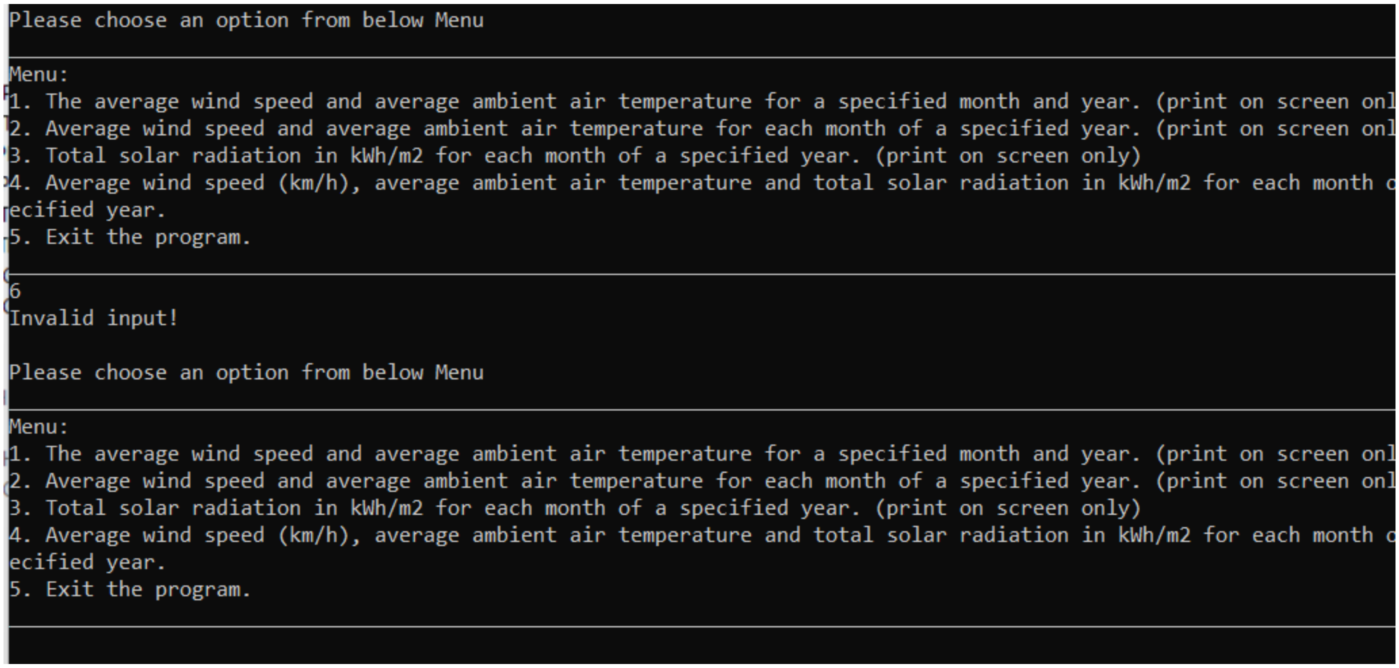
```
1993
No Data

```

Test ID 9:



```
Menu:
1. The average wind speed and average ambient air temperature for a specified month and year. (print on screen only)
2. Average wind speed and average ambient air temperature for each month of a specified year. (print on screen only)
3. Total solar radiation in kWh/m2 for each month of a specified year. (print on screen only)
4. Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month of a sp
ecified year.
5. Exit the program.

5
Exiting program!

Process returned 0 (0x0)   execution time : 228.091 s
Press any key to continue.
```

Test ID 10:



```
Please choose an option from below Menu

Menu:
1. The average wind speed and average ambient air temperature for a specified month and year. (print on screen onl
2. Average wind speed and average ambient air temperature for each month of a specified year. (print on screen onl
3. Total solar radiation in kWh/m2 for each month of a specified year. (print on screen only)
4. Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month o
ecified year.
5. Exit the program.

6
Invalid input!

Please choose an option from below Menu

Menu:
1. The average wind speed and average ambient air temperature for a specified month and year. (print on screen onl
2. Average wind speed and average ambient air temperature for each month of a specified year. (print on screen onl
3. Total solar radiation in kWh/m2 for each month of a specified year. (print on screen only)
4. Average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month o
ecified year.
5. Exit the program.
```