MODULE NAME:	MODULE CODE:
PROGRAMMING 2A	PROG6211
PROGRAMMING 2A	PROG6221

ASSESSMENT TYPE: POE (PAPER)

TOTAL MARK ALLOCATION: 100 MARKS

TOTAL HOURS: A minimum of 15 HOURS is suggested to complete this assessment

By submitting this assignment, you acknowledge that you have read and understood all the rules as per the terms in the registration contract, in particular the assignment and assessment rules in The IIE Assessment Strategy and Policy (IIE009), the intellectual integrity and plagiarism rules in the Intellectual Integrity Policy (IIE023), as well as any rules and regulations published in the student portal.

INSTRUCTIONS:

1. ***No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks. No more than 10% of the assignment may consist of direct quotes.***
2. ***Any assignment with a similarity index of more than 25% will be scrutinised for plagiarism. Please make sure you attach a similarity report to your POE if required.***
3. ***Make a copy of your assignment before handing it in.***
4. *Assignments must be typed unless otherwise specified.*
5. *All work must be adequately and correctly referenced.*
6. *Begin each section on a new page.*
7. *Follow all instructions on the assignment cover sheet.*
8. *This is an individual assignment.*

Referencing Rubric

Providing evidence based on valid and referenced academic sources is a fundamental educational principle and the cornerstone of high-quality academic work. Hence, The IIE considers it essential to develop the referencing skills of our students in our commitment to achieve high academic standards. Part of achieving these high standards is referencing in a way that is consistent, technically correct and congruent. This is not plagiarism, which is handled differently.

Poor quality formatting in your referencing will result in a penalty **of a maximum of ten percent being deducted from the mark awarded**, according to the following guidelines. Please note, however, that **evidence of plagiarism in the form of copied or uncited work (not referenced), absent reference lists, or exceptionally poor referencing, may result in action being taken in accordance with The IIE's Intellectual Integrity Policy (0023)**.

Markers are required to provide feedback to students by indicating (circling/underlining) the information that best describes the student's work.

Minor technical referencing errors: 5% deduction from the overall mark – the student's work contains five or more errors listed in the minor errors column in the table below.

Major technical referencing errors: 10% deduction from the overall mark – the student's work contains five or more errors listed in the major errors column in the table below.

If both minor and major errors are indicated, then 10% only (and not 5% or 15%) is deducted from the overall mark.

The examples provided below are not exhaustive but are provided to illustrate the error.

Required: **Technically correct referencing style**	**Minor errors** **in technical correctness of referencing style** **Deduct 5% from mark awarded**	**Major errors** **In technical correctness of referencing style** **Deduct 10% from mark awarded**
Consistency The same referencing format has been used for all in-text references and in the bibliography/reference list.	Minor inconsistencies. The referencing style is generally consistent, but there are one or two changes in the format of in-text referencing and/or in the bibliography. For example, page numbers for direct quotes (in-text) have been provided for one source, but not in another instance. Two book chapters (bibliography) have been referenced in the bibliography in two different formats.	Major inconsistencies. Poor and inconsistent referencing style used in-text and/or in the bibliography/ reference list. Multiple formats for the same type of referencing have been used. For example, the format for direct quotes (in-text) and/or book chapters (bibliography/ reference list) is different across multiple instances.
Technical correctness Referencing format is technically correct throughout the submission. Position of the reference: a reference is directly associated with every concept or idea. For example, quotation marks, page numbers, years, etc. are applied correctly, sources in the bibliography/reference list are correctly presented.	Generally, technically correct with some minor errors. The correct referencing format has been consistently used, but there are one or two errors. Concepts and ideas are typically referenced, but a reference is missing from one small section of the work. Position of the references: references are only given at the beginning or end of every paragraph. For example, the student has incorrectly presented direct quotes (in-text) and/or book chapters (bibliography/reference list).	Technically incorrect. The referencing format is incorrect. Concepts and ideas are typically referenced, but a reference is missing from small sections of the work. Position of the references: references are only given at the beginning or end of large sections of work. For example, incorrect author information is provided, no year of publication is provided, quotation marks and/or page numbers for direct quotes missing, page numbers are provided for paraphrased material, the incorrect punctuation is used (in-text); the bibliography/reference list is not in alphabetical order, the incorrect format for a book chapter/journal article is used, information is missing e.g. no place of publication had been provided (bibliography); repeated sources on the reference list.
Congruence between in-text referencing and bibliography/reference list All sources are accurately reflected and are all accurately included in the bibliography/reference list.	Generally, congruence between the in-text referencing and the bibliography/ reference list with one or two errors. There is largely a match between the sources presented in-text and the bibliography. For example, a source appears in the text, but not in the bibliography/reference list or vice versa.	A lack of congruence between the in-text referencing and the bibliography. No relationship/several incongruencies between the in-text referencing and the bibliography/reference list. For example, sources are included in-text, but not in the bibliography and vice versa, a link, rather than the actual reference is provided in the bibliography.
In summary: the recording of references is accurate and complete.	In summary, at least 80% of the sources are correctly reflected and included in a reference list.	In summary, at least 60% of the sources are incorrectly reflected and/or not included in reference list.

Overall Feedback about the consistency, technical correctness and congruence between in-text referencing and bibliography:

_____

_____

Portfolio of Evidence (PoE) — Background

The Portfolio of Evidence (PoE) requires you to create a weather forecasting system. The application will progressively be developed through the execution of Task 1 and 2 that build on each other to create the final application for PROG6211. As part of the final PoE submission you need to adjust the application that you developed for Task 1 and Task 2 by implementing database interaction and developing a web application.

Summary Sheet:

ITEM	DESCRIPTION
Summary of Activities	The student needs to submit online.
Tools & Resources	• Access to the internet for online submission; • Visual Studio .NET.
Additional Information	• **Task 1:** Final completed Task 1 must be submitted after LU3. • **Task 2:** Final completed Task 2 must be submitted after LU4. • **PoE:** Final completed PoE must be submitted after LU5.

By completing Tasks 1 and 2 in this Portfolio of Evidence you will develop a weather forecasting system that will be used by both weather forecasters and general users.

The following functionality should be available to weather forecasters:

- Capture weather forecasts (specify city, date, minimum temperature, maximum temperature, precipitation, humidity, and wind speed. For example:

PROMPT	INPUT BY WEATHER FORECASTER
City	Cape Town
Date	1 January 2019
Minimum Temperature	20°C
Maximum Temperature	27°C
Precipitation	60%
Humidity	74%
Wind speed	18 km/h

- View and edit weather forecasts.

- View the following weather forecast reports for specific date ranges:
    1) Report indicating weather forecasts for a specific city within a specific date range. This report should indicate the minimum temperature, maximum temperature, precipitation, humidity, and wind speed per day, as well as highlight the lowest minimum temperature, as well as the highest maximum temperature, precipitation, humidity, and wind speed that was forecasted within the specified date range.
    2) Report comparing weather forecasts for selected cities within a specific date range. This report should indicate the minimum temperature, maximum temperature, precipitation, humidity, and wind speed per day for each of the selected cities. For each day the lowest minimum temperature, as well as the highest maximum temperature, precipitation, humidity, and wind speed that was forecasted for the selected cities should be highlighted.

The following functionality should be available to general users:

- Specify cities that they regularly want to view weather forecasts for.
- View weather forecasts for the specified cities.

Ensure that you have covered all items listed in the marking rubric in Appendix A.

Task 1 — Weather Forecasting System (Marks: 100)

Learning Units: LU1 – 3

Assessment:

Assessment/Deliverable	Marks	Weight	Duration
Task 1	100	25%	15hrs

Consider the description of the weather forecasting system as described on the previous page and develop a **Windows Forms application** with the following functionality described below. All data should appropriately be captured in **arrays** and other relevant **variables**.

The application should allow weather forecasters to:

- Capture weather forecasts by specifying the city, date, minimum temperature, maximum temperature, precipitation, humidity, and wind speed.
- Generate a weather forecast report for a specific city within a specified date range: The weather forecaster should select a city, start date and end date and then a report indicating the minimum temperature, maximum temperature, precipitation, humidity, and wind speed per day should be generated and displayed. The lowest minimum temperature, as well as the highest maximum temperature, precipitation, humidity, and wind speed that was forecasted within the specified date range should be highlighted.
- Generate a weather forecast report for multiple cities within a specific date range : The weather forecaster should select all the cities that should be included in the report, as well as specify the start and end date that the report should be based on. The minimum temperature, maximum temperature, precipitation, humidity, and wind speed per day, for each of the selected cities should then be displayed. For each day the lowest minimum temperature, as well as the highest maximum temperature, precipitation, humidity, and wind speed that was forecasted for the selected cities should also be highlighted.

Task 2 — Advanced Weather Forecasting System (Marks: 100)

Learning Unit: LU4

Assessment:

Assessment/Deliverable	Marks	Weight	Duration
Task 2	100	30%	15hrs

Adjust the weather forecasting application developed in Task 1 to appropriately store all data in **files**. For weather forecasts the following should be stored in files:

- City;
- Date;
- Minimum temperature;
- Maximum temperature;
- Precipitation;
- Humidity;
- Wind speed.

Also make use of files to store login details for weather forecasters and general users.

Once the necessary changes have been made to store all data in files, add the following functionality for weather forecasters:

- Allow weather forecasters to log in before capturing weather forecasts and viewing reports.
- Allow weather forecasters to view and edit captured weather forecasts.
- Reports should be written to file in order to allow the weather forecaster to print out the reports.

Also add the necessary functionality to allow regular users to log in to use the system. Regular users should not have access to the functionality available to weather forecasters. General users should be able to:

- Specify cities that they regularly want to view weather forecasts for.
- View weather forecasts for the specified cities.

PoE — Advanced Weather Forecasting System and Web-based Weather Viewer (Marks: 100)

Learning Unit: LU5

Assessment:

Assessment/Deliverable	Marks	Weight	Duration
POE	100	35%	15hrs

Adjust the weather forecasting application developed in Task 1 and Task 2 to appropriately store all data in a **database**. For weather forecasts the following should be stored in the database:

- City;
- Date;
- Minimum temperature;
- Maximum temperature;
- Precipitation;
- Humidity;
- Wind speed.

The weather forecaster should be able to edit all of the data above.

Also, adjust the application to store all login details in the database. When general users specify cities that they regularly want to view weather forecasts for it should also be stored in the database.

In addition to the weather forecasting application developed in Task 1 and Task 2 and adjusted above, also develop a **web-based application** that allows general users to log in and view the latest weather forecast for all of the cities that they specified.

Appendix A

Assessment Sheet (Marking Rubric)

MODULE NAME:	MODULE CODE:
PROGRAMMING 2A	PROG6211
PROGRAMMING 2A	PROG6221

STUDENT NAME:
STUDENT NUMBER:

RUBRIC 1 (for Task 1) — OUTLINE	Levels of Achievement				Feedback
In order to be awarded full marks for these elements of Task 1, students need to have:	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Good coding standards: Comments/code readability throughout C# code.	4	3	1—2	0	
Good coding standards: C# file naming/modular coding — using different C# files for different modules.	4	3	1—2	0	
Good coding standards: Efficient code (no redundancy).	4	3	1—2	0	
User interface: Appropriate controls employed to support functionality.	4	3	1—2	0	
User interface: Appropriate event handling to support functionality.	4	3	1—2	0	

RUBRIC 1 (for Task 1) — OUTLINE [continued]	Levels of Achievement				Feedback
In order to be awarded full marks for these elements of Task 1, students need to have:	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Application functionality: All data appropriately captured in relevant arrays and variables.	**8—10**	**5—7**	**3—4**	**0—2**	
Application functionality: Weather forecaster able to capture weather forecasts.	**8—10**	**5—7**	**3—4**	**0—2**	
Application functionality: Weather forecaster able to generate weather forecast report for a specific city within a specified date range. Lowest minimum, as well as the highest maximum temperature, precipitation, humidity, and wind speed appropriately highlighted.	**21—30**	**13—21**	**7—12**	**0—6**	
Application functionality: Weather forecaster able to generate weather forecast report for multiple cities within a specified date range. For each day the lowest minimum temperature, as well as the highest maximum temperature, precipitation, humidity, and wind speed that was forecasted for the selected cities appropriately highlighted.	**21—30**	**13—21**	**7—12**	**0—6**	
TASK 1 TOTAL					**/100**

MODULE NAME:	MODULE CODE:
PROGRAMMING 2A	PROG6211
PROGRAMMING 2A	PROG6221

STUDENT NAME:
STUDENT NUMBER:

RUBRIC 2 (for Task 2) — OUTLINE	Levels of Achievement				Feedback
In order to be awarded full marks for these elements of Task 2, students need to have:	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Good coding standards: Comments/code readability throughout C# code.	4	3	1—2	0	
Good coding standards: C# file naming/modular coding — using different C# files for different modules.	4	3	1—2	0	
Good coding standards: Efficient code (no redundancy).	4	3	1—2	0	
User interface: Appropriate controls employed to support functionality.	4	3	1—2	0	
User interface: Appropriate event handling to support functionality.	4	3	1—2	0	
Application functionality: All data appropriately stored in files.	15—20	9—14	5—8	0—4	
Application functionality: Weather forecasters and general users able to log in.	8—10	5—7	3—4	0—2	

RUBRIC 2 (for Task 2) — OUTLINE [continued]	Levels of Achievement				Feedback
In order to be awarded full marks for these elements of Task 2, students need to have:	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Application functionality: Appropriate user interface depending on whether it is weather forecaster or general user logging in.	**8—10**	**5—7**	**3—4**	**0—2**	
Application functionality: Weather forecaster able to view and edit weather forecasts.	**8—10**	**5—7**	**3—4**	**0—2**	
Application functionality: Reports appropriately written to file. Data written in neat columns, with appropriate headings where relevant.	**8—10**	**5—7**	**3—4**	**0—2**	
Application functionality: General user able to specify cities that they want to regularly view weather forecasts for.	**8—10**	**5—7**	**3—4**	**0—2**	
Application functionality: General user able to view weather forecasts for the specified cities.	**8—10**	**5—7**	**3—4**	**0—2**	
TASK 2 TOTAL					**/100**

MODULE NAME:	MODULE CODE:
PROGRAMMING 2A	PROG6211
PROGRAMMING 2A	PROG6221

STUDENT NAME:
STUDENT NUMBER:

RUBRIC 3 (for POE) — OUTLINE	Levels of Achievement				Feedback
In order to be awarded full marks for these elements of the PoE, students need to have:	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Good coding standards: Comments/code readability throughout C# code.	4	3	1—2	0	
Good coding standards: C# file naming/modular coding — using different C# files for different modules.	4	3	1—2	0	
Good coding standards: Efficient code (no redundancy).	4	3	1—2	0	
User interface: Appropriate controls employed to support functionality.	4	3	1—2	0	
User interface: Appropriate event handling to support functionality.	4	3	1—2	0	
Application functionality: All data appropriately stored in database.	15—20	9—14	5—8	0—4	

RUBRIC 3 (for PoE) — OUTLINE [continued]	Levels of Achievement				Feedback
In order to be awarded full marks for these elements of the PoE, students need to have:	Excellent	Good	Developing	Poor	
	Score Ranges Per Level (½ marks possible)				
Application functionality: General users able to log in via web-based application.	**15—20**	**9—14**	**5—8**	**0—4**	
Application functionality: General users able to select city from their list of favourite cities that they want to view weather forecast for via the web-based application.	**15—20**	**9—14**	**5—8**	**0—4**	
Application functionality: General user able to view weather forecast for selected city via the web-based application.	**15—20**	**9—14**	**5—8**	**0—4**	
PoE TOTAL					**/100**