

COMP1811 – Python Project Report

Name		Student ID	
------	--	------------	--

1. BRIEF STATEMENT OF FEATURES YOU HAVE COMPLETED

1.1 Circle the parts of the coursework you have fully completed and are fully working . Please be accurate.	Features F1: i <input type="checkbox"/> ii <input type="checkbox"/> iii <input type="checkbox"/> iv <input type="checkbox"/> F2: i <input type="checkbox"/> ii <input type="checkbox"/> F3: i <input type="checkbox"/> ii <input type="checkbox"/>
1.2 Circle the parts of the coursework you have partly completed or are partly working .	Features F1: i <input type="checkbox"/> ii <input type="checkbox"/> iii <input type="checkbox"/> iv <input type="checkbox"/> F2: i <input type="checkbox"/> ii <input type="checkbox"/> F3: i <input type="checkbox"/> ii <input type="checkbox"/>
Briefly explain your answer if you circled any parts in 1.2	

2. CONCISE LIST OF BUGS AND WEAKNESSES

*A concise list of bugs and/or weaknesses in your work (if you don't think there are any, then say so). Bugs that are declared in this list will lose you fewer marks than ones that you don't declare! (**100-200 word**, but word count depends heavily on the number of bugs and weaknesses identified.)*

2.1 BUGS

List each bug plus a brief description

2.2 WEAKNESSES

List each weakness plus a brief description

3.

3. DESCRIPTION OF THE FEATURES IMPLEMENTED

Describe your implementation of the required features and how well do they work. Provide some exposition of the design decisions made and indicate how the features developed were integrated.

4. CLASSES AND OOP FEATURES

*List all the classes used in your program and include the attributes and behaviours for each. You may use a class diagram to illustrate these classes. Your narrative for section 3.2 should describe the design decisions you made and the OOP techniques used. List the classes you developed and provide an exposition on the choice of classes, class design and OOP features implemented. (**200-400 words**).*

4.1 CLASSES USED

4.2 BRIEF EXPLANATION OF CLASS DESIGN AND OOP FEATURES USED

5. CODE FOR THE CLASSES CREATED

*Add the **code for each of the classes you have implemented yourself** here. If you have contributed to parts of classes, please highlight those parts in a different colour. Copy and paste relevant code - actual code please, no screenshots! Make it easy for the tutor to read. Add explanation if necessary – though your in-code comments should be clear enough. You will lose marks if screenshots are provided instead of code.*

(DO NOT provide a listing of the entire code. You will be marked down if a full code listing is provided.)

5.1 CLASS ...

5.2 CLASS ...

5.3 CLASS ...

...

6. TESTING

Describe the process you took to test your code and to make sure the program functions as required. Provide the detailed test plan used.

7. ANNOTATED SCREENSHOTS DEMONSTRATING IMPLEMENTATION

*Provide screenshots that demonstrate the features implemented. Annotate each screenshot and if necessary, provide a brief description for **each (up to 100 words)** to explain the code in action. Make sure the screenshots make clear what you have implemented and achieved.*

7.1 FEATURE F1

A. SUB-FEATURE I- SCREENSHOTS ...

B. SUB-FEATURE II- SCREENSHOTS ...

C. SUB-FEATURE III- SCREENSHOTS ...

D. SUB-FEATURE IV SCREENSHOTS ...

7.2 FEATURE F2

A. SUB-FEATURE I- SCREENSHOTS ...

B. SUB-FEATURE II- SCREENSHOTS ...

7.3 FEATURE F3

A. SUB-FEATURE I- SCREENSHOTS ...

B. SUB-FEATURE II- SCREENSHOTS ...

8. EVALUATION

*Give a reflective, critical self-evaluation of your experience developing the project and discuss what you would do if you had more time to work on the project. Answer the following questions for the reflection and write **350-400 words overall**. Please include an actual word count for this section.*

8.1 EVALUATE HOW WELL YOUR DESIGN AND IMPLEMENTATION MEET THE REQUIREMENTS

8.2 EVALUATE YOU OWN PERFORMANCE

8.2.1 WHAT WENT WELL?

8.2.2 WHAT WENT LESS WELL?

8.2.3 WHAT WAS LEARNT?

8.2.4 HOW WOULD A SIMILAR TASK BE COMPLETED DIFFERENTLY?

8.2.5 HOW COULD THE MODULE BE IMPROVED?

8.3 SELF-ASSESSMENT

Please assess yourself objectively for each section shown below and then enter the total mark you expect to get. Marks for each assessment criteria are indicated between parentheses.

CODE DEVELOPMENT (70)

Features Implemented [40]

Feature 1 (up to 16)

- Sub-features have not been implemented – 0
- Attempted, not complete or very buggy – 1 to 5
- Implemented and functioning without errors but not integrated – 6 to 9
- Implemented and fully integrated but buggy – 10 to 14
- Implemented, fully integrated and functioning without errors – 15 to 16

Feature 2 (up to 16)

- Sub-features have not been implemented – 0
- Attempted, not complete or very buggy – 1 to 5
- Implemented and functioning without errors but not integrated – 6 to 9
- Implemented and fully integrated but buggy – 10 to 14
- Implemented, fully integrated and functioning without errors – 15 to 16

Feature 3 (up to 8)

- Sub-features have not been implemented – 0
- Attempted, not complete or very buggy – 1 to 3
- Implemented and functioning without errors but not integrated – 4 to 5
- Implemented and fully integrated but buggy – 6 to 7
- Implemented, fully integrated and functioning without errors – 8

For this criterion I think I got: out of 40

Use of OOP techniques [20]

Abstraction (up to 8)

- No classes have been created – 0
- Classes have been created superficially and not instantiated or used – 1 or 2

Classes have been created but only some have been instantiated and used – 3 or 4

Useful classes and objects have been created and used correctly – 5 or 6

The use of classes and objects exceeds the specification – 7 or 8

Encapsulation (up to 8)

No encapsulation has been used – 0

Class variables and methods have been encapsulated superficially – 1 to 3

Class variables and methods have been encapsulated correctly – 4 to 6

The use of encapsulation exceeds the specification – 7 to 8

Inheritance (up to 4)

No inheritance has been used – 0

Classes have been inherited superficially – 1 or 2

Classes have been inherited correctly – 3

The use of inheritance exceeds the specification – 4

Bonus marks will be awarded for the appropriate use of polymorphism (bonus marks up to 5)

For this criterion I think I got: out of 20

Quality of Code [10]

Code Duplication (up to 4)

Code contains too many unnecessary code repetition – 0

Regular occurrences of duplicate code – 1

Occasional duplicate code – 2

Very little duplicate code – 3

No duplicate code – 4

PEP8 Conventions and naming of variables, methods and classes (up to 3)

PEP8 and naming convention has not been used – 0

PEP8 and naming convention has been used occasionally – 1

PEP8 and naming convention has been used, but not regularly – 2

PEP8 convention used professionally and all items have been named correctly – 4

In-code Comments (up to 3)

No in-code comments – 0

Code contains occasional in-code comments – 1

Code contains useful and regular in-code comments – 2

Thoroughly commented, good use of docstrings, and header comments describing.py files – 3

For this criterion I think I got: out of 10

DOCUMENTATION (20)

Design (up to 10) clear exposition about the design and decisions for OOP use

The documentation cannot be understood on first reading or is mostly incomplete – 0

The documentation is readable, but a section(s) are missing – 1 to 3

The documentation is complete – 4 to 6

The documentation is complete and of a high standard – 7 to 10

Testing (5)

Testing has not been demonstrated in the documentation – 0

Little white box testing has been documented – 1 or 2

White box testing has been documented for all the coursework – 3 or 4

White box testing has been documented for the whole system – 5

Evaluation (5)

No evaluation was shown in the documentation – 0

The evaluation shows a lack of thought – 1 or 2

The evaluation shows thought – 3 or 4

The evaluation shows clear introspection, demonstrates increased awareness – 5

For this criterion I think I got: out of 20

ACCEPTANCE TESTS - SCREENCAST (10)

Screencast (up to 10)

Not submitted or no work demonstrated – 0

Work demonstrated was not up to the standard expected – 1 to 3

Work demonstrated was up to the standard expected – 4 to 7

Work demonstrated exceeded the standard expected – 8 to 10

For this criterion I think I got: out of 10

I think my overall mark would be: out of 100

APPENDIX A: CODE LISTING

*Provide a complete listing of all the *.py files in your PyCharm project. Make sure your code is well commented and applies professional Python convention (refer to [PEP 8](#) for details). The code listed here must match that uploaded to Moodle. Please copy and paste the actual code – no screenshots please! You will lose marks if screenshots are provided instead of code.*