Assignment 2

Submission Date: 17th December March 2024 @ 2330

Weighting: 70%

WARNING:

- All submitted work will be electronically scanned for plagiarism and the use of Artificial Intelligence (AI) software. The work that you submit must be your own; any material from other sources must be correctly cited and referenced. If unreferenced material is detected a student will be reported for plagiarism. AI should not be used in the production of this work. Use of AI which has not been acknowledged is an academic misconduct offence and again will lead to a report of plagiarism.
- Assessor reserve the right to viva you to explain the code and it's implementation.

Learning Outcomes

On completion of the assignment, the student will develop:

1. an in-depth understanding of object-oriented programming
2. understanding of the basic principles of mobile apps design
3. skills in designing a mobile app.

Brief

1. **Expansion of Lab Application:**

   The last UK goverment tried to bring back imperial units over metric (EU rules). However that government suffered a humilating defeat as 100k traders rejected the change. If this change had happened the UK would have to convert to imperial units, with that in mind, you are to expand on the Mass Converter App to convert between 12 different imperial units overleaf.

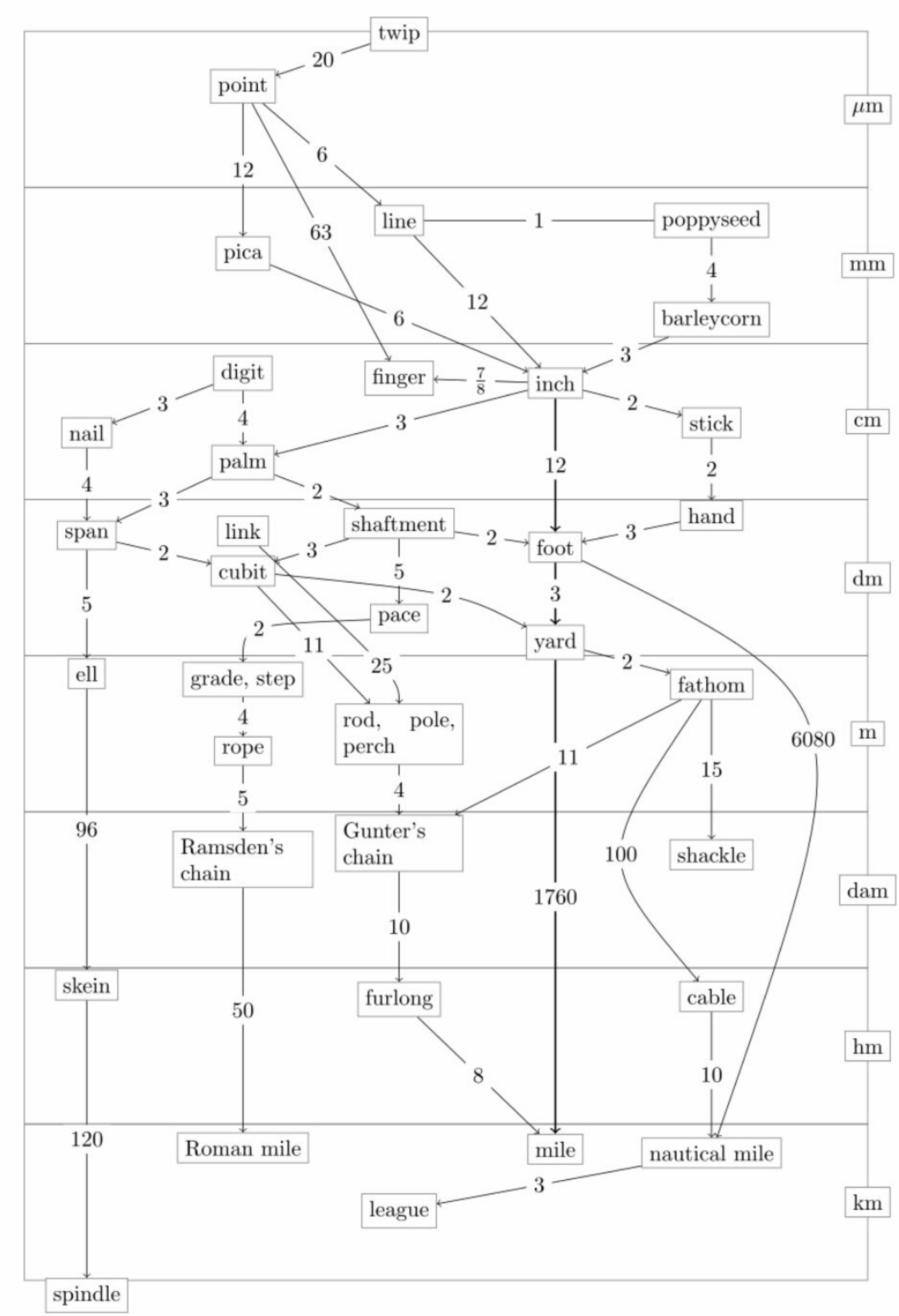
2. **Implementation Requirements:**

   - Utilise OOP principles to define and manage data structures and classes.
   - Ensure the application follows clean code practices, including proper naming conventions, modularity, and efficient use of resources.
   - Implement data validation checks to ensure the accuracy and integrity of the data.

3. **Documentation:**

   - Use KDoc to document classes, methods, and important code segments.
   - Provide a comprehensive README file explaining the application's features, design choices, and usage instructions.

Continued overleaf


twip
20
point
μm
12
6
63
line
1
poppyseed
pica
mm
4
12
6
barleycorn
3
digit
finger
7/8
inch
3
4
3
2
nail
stick
cm
palm
12
2
4
3
2
hand
span
link
shaftment
2
foot
3
2
3
cubit
5
3
dm
5
2
pace
2
yard
11
25
2
ell
grade, step
fathom
4
rod, pole, perch
rope
11
15
6080
m
5
4
96
Gunter's chain
Ramsden's chain
100
shackle
dam
1760
10
skein
50
furlong
cable
hm
8
10
120
Roman mile
mile
nautical mile
3
league
km
spindle

Coursework Submission Link

- Moodle submission Engineers
- Moodle submission Technologists
- **Source Code:** Submit the complete source code of the application, including all necessary files to compile and run the project.
- **Documentation:** Include:
  - KDoc comments within the code.
  - README explaining the added features, design choices, and how OOP principles were applied
- **Executable APK:** Provide an executable APK file of the final application.

Rubic Summary

Full rubric is at the bottom of the brief.

1. **OOP Implementation (25%)**
   - Effective use of classes and objects.
   - Proper implementation of inheritance, polymorphism, encapsulation, and abstraction.
   - Efficient data management and manipulation.
2. **Clean Code (25%)**
   - Adherence to clean code principles.
   - Readability and maintainability of the code.
   - Proper naming conventions, modularity, and code organization.
3. **KDoc Utilisation (10%)**
   - Comprehensive and clear documentation using KDoc.
   - Proper use of comments to explain complex code segments.
   - Inclusion of usage examples where appropriate.
4. **README (15%)**
   - Correct syntax
   - Explanation of design choice
   - explanation of the application's features
   - How to use the application
5. **App Design (25%)**
   - Quality and intuitiveness of the user interface (UI).
   - User experience (UX) enhancements.
   - Aesthetic design and visual appeal.
   - Functionality and usability of new features.

Continued overleaf

Rubric in Full

Clean Code

Performance Level	Description
0-29% Fail	Code exhibits a lack of understanding and application of clean code principles. The structure is severely deficient, hindering comprehension. Naming conventions are inconsistent and fail to adhere to best practices. The overall code quality is substantially below the expected standard.
30-39% Fail	Limited understanding of clean code principles is evident. The code structure is basic, but significant improvements are needed for clarity. Naming conventions are inconsistent and do not fully adhere to best practices. The code, while functional, lacks the sophistication expected in clean code.
40-49% Satisfactory	The code demonstrates a basic understanding of clean code principles. The structure meets minimum requirements but lacks sophistication. Naming conventions are basic and partially follow best practices. While the code is acceptable, it falls short of achieving a higher level of cleanliness and clarity.
50-59% Good	There is a good understanding and application of clean code principles. The code structure is above average, contributing to clarity and understanding. Naming conventions are good and mostly follow best practices. The code demonstrates a commendable effort in adhering to clean code standards.
60-69% Very Good	The code exhibits a very good understanding of clean code principles. The structure is impressive, contributing to a high level of clarity and understanding. Naming conventions are very good and consistently follow best practices. The code showcases a level of cleanliness that goes beyond the expected standard.
70-79% Excellent	An excellent understanding and application of clean code principles are evident. The code structure is exceptional, setting a benchmark for clarity and understanding. Naming conventions are excellent and consistently follow best practices. The code exemplifies an exceptional standard of cleanliness and organization.
80-100% Exceptional	The code demonstrates an exceptional understanding and application of clean code principles. The structure is outstanding, demonstrating an unparalleled level of clarity and understanding. Naming conventions are exceptional and consistently follow best practices. The code sets an extraordinary standard for cleanliness and represents an outstanding example of clean and maintainable code.

Continued overleaf

OOP

Performance Level	Description
0-29% Fail	Inadequate use of classes and objects. Lack of implementation of key OOP principles. Data management is inefficient and poorly structured.
30-39% Fail	Basic use of classes and objects with significant errors. Limited implementation of OOP principles. Data management needs major improvement.
40-49% Satisfactory	Basic understanding of OOP principles. Adequate use of classes and objects. Data management is functional but lacks sophistication.
50-59% Good	Good understanding of OOP principles. Effective use of classes and objects. Data management is above average but not exceptional.
60-69% Very Good	Very good understanding of OOP principles. Well-structured use of classes and objects. Data management is efficient and well-organized.
70-79% Excellent	Excellent understanding of OOP principles. Advanced use of classes and objects. Data management is highly efficient and structured.
80-100% Exceptional	Outstanding understanding of OOP principles. Exceptional use of classes and objects. Data management is exemplary and sets a high standard.

Continued overleaf

KDoc

Performance Level	Description
0-29% Fail	Lack of KDoc documentation. Minimal or no comments explaining the code. No usage examples provided.
30-39% Fail	Limited KDoc documentation. Few comments explaining the code. Usage examples are incomplete or missing.
40-49% Satisfactory	Basic KDoc documentation. Some comments explaining the code. Usage examples are basic and incomplete.
50-59% Good	Good KDoc documentation. Adequate comments explaining the code. Usage examples are helpful but not comprehensive.
60-69% Very Good	Very good KDoc documentation. Clear and helpful comments explaining the code. Usage examples are detailed and informative.
70-79% Excellent	Excellent KDoc documentation. Comments are clear, detailed, and enhance understanding. Usage examples are comprehensive.
80-100% Exceptional	Exceptional KDoc documentation. Comments are exceptionally clear and detailed. Usage examples are outstanding and greatly enhance understanding.

Continued overleaf

README

Performance Level	Description
0-29% Fail	README is missing or severely lacking in detail. Syntax is incorrect. Design choices and application features are not explained. Usage instructions are unclear or missing.
30-39% Fail	README is incomplete. Syntax is largely incorrect. Design choices and application features are poorly explained. Usage instructions are minimal and unclear.
40-49% Satisfactory	README is basic but functional. Syntax is partially correct. Design choices and application features are briefly explained. Usage instructions are present but not detailed.
50-59% Good	README is good and functional. Syntax is mostly correct. Design choices and application features are adequately explained. Usage instructions are clear but could be more detailed.
60-69% Very Good	README is very good and detailed. Syntax is correct. Design choices and application features are well explained. Usage instructions are clear and detailed.
70-79% Excellent	README is excellent and comprehensive. Syntax is correct. Design choices and application features are thoroughly explained. Usage instructions are clear, detailed, and easy to follow.
80-100% Exceptional	README is exceptional in detail and clarity. Syntax is flawless. Design choices and application features are excellently explained. Usage instructions are outstandingly clear, detailed, and easy to follow.

Continued overleaf

App Design/Usability

Performance Level	Description
0-29% Fail	User interface is poorly designed and unintuitive. User experience is severely lacking. Aesthetic design is unappealing. Functionality and usability of new features are minimal or non-existent. Few or no significant features added.
30-39% Fail	User interface is basic and not intuitive. User experience is limited. Aesthetic design needs significant improvement. Functionality and usability of new features are limited. Less than three significant features added, and those added lack meaningful impact.
40-49% Satisfactory	User interface is functional but lacks sophistication. User experience is basic. Aesthetic design is acceptable but could be improved. Functionality and usability of new features are satisfactory. Three significant features added, but their implementation lacks depth or polish.
50-59% Good	User interface is good and fairly intuitive. User experience is positive. Aesthetic design is good but not outstanding. Functionality and usability of new features are above average. Three significant features added and implemented competently.
60-69% Very Good	User interface is very good and intuitive. User experience is very positive. Aesthetic design is visually appealing. Functionality and usability of new features are very good. Three significant features added with a high level of functionality and integration.
70-79% Excellent	User interface is excellent and highly intuitive. User experience is excellent. Aesthetic design is highly appealing. Functionality and usability of new features are excellent. Three significant features added with excellent functionality and seamless integration.
80-100% Exceptional	User interface is outstanding and exceptionally intuitive. User experience is exceptional. Aesthetic design is outstanding and visually striking. Functionality and usability of new features are exceptional and innovative. Three or more significant features added with outstanding functionality and innovative implementation.

End of Brief