The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, AUTUMN SEMESTER 2020-2021

DEVELOPING MAINTAINABLE SOFTWARE (COMP2013)

Maximum Time Allowed to Submit Answers in Moodle: 3 HOURS

This is a take-home and open-book exam with answers to be submitted in Moodle no later than the DATE/TIME indicated in the Moodle dropbox. Submissions after such deadline will not be accepted. The maximum time allowed already includes additional time for IT issues and upload. For students with approved support plans, additional time will be set accordingly in the Moodle dropbox.

Answer ALL 15 QUESTIONS

Submit your answers containing all the work you wish to have marked as a single PDF file, with each page in the correct orientation, to the appropriate dropbox in the module's Moodle page.

You may write/draw your answers on paper and then scan them to a PDF file, or you may type/draw your answers into electronic form directly and generate a PDF file. Guidance on scanning can be found through the Faculty of Science Moodle Page Guidance for Remote Learning.

Your solutions should include complete explanations and should be based on the material covered in the module. Make sure your PDF file is easily readable and does not require magnification. Text/drawing which is not in focus or is not legible for any other reason will be ignored.

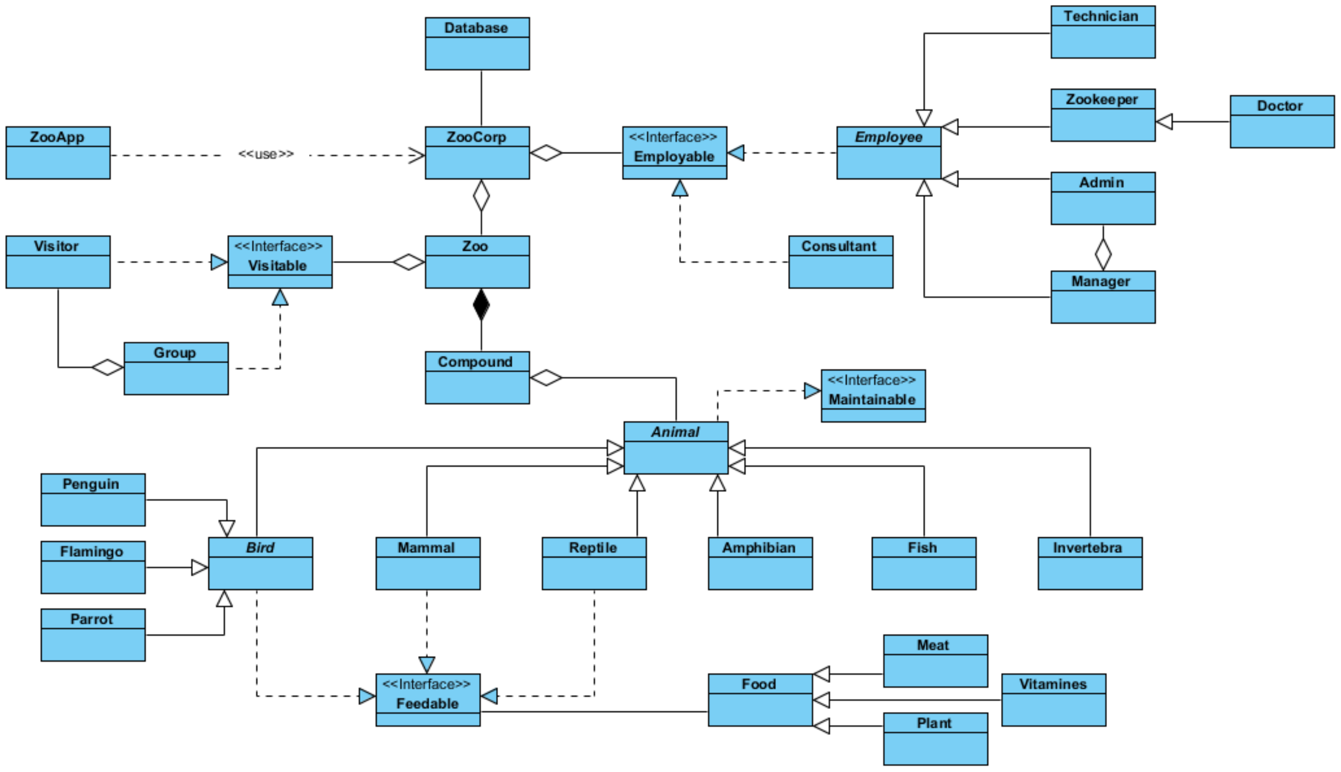
Use the following naming convention for your PDF file: StudentID_COMP2013. Include your student ID number at the top of each page of your answers. Do not include your name.

Staff are not permitted to answer assessment or teaching queries during the period in which your examination is live. If you spot what you think may be an error on the exam paper, note this in your submission but answer the question as written.

You must produce the answers by yourself only. You must be careful to avoid plagiarism, collusion or false authorship. Please familiarise yourself with the Faculty of Science Statement on Academic Integrity. This statement refers to, and does not replace, the University policy which stipulates severe penalties for academic misconduct. Please check the box indicated on Moodle to confirm that you have read this statement and that you understand it.

Note: You can answer the questions in any order, there are no dependencies between them. Have a quick look through the exam before you start.

When we talk about the **ZooProject** in the questions we always relate to the project presented in the following diagram:



Question 1. Developing Maintainable Software:

a. Briefly explain the relationship between writing maintainable software and software maintenance. [5 marks]
b. Illustrate your answer from (a) with an example that relates to the **ZooProject**. [5 marks]
c. Provide a practical examples of how you could add "robustness" to the **ZooProject** code, leading to software that is easy to maintain and extend, and justify why your example would make the code maintainable. [7 marks]
d. Provide a second practical examples of how you could add "robustness" to the **ZooProject** code, leading to software that is easy to maintain and extend, and justify why your example would make the code maintainable. [7 marks]

[Allowed maximum length of your answer: 250 words]

[overall 24 marks]

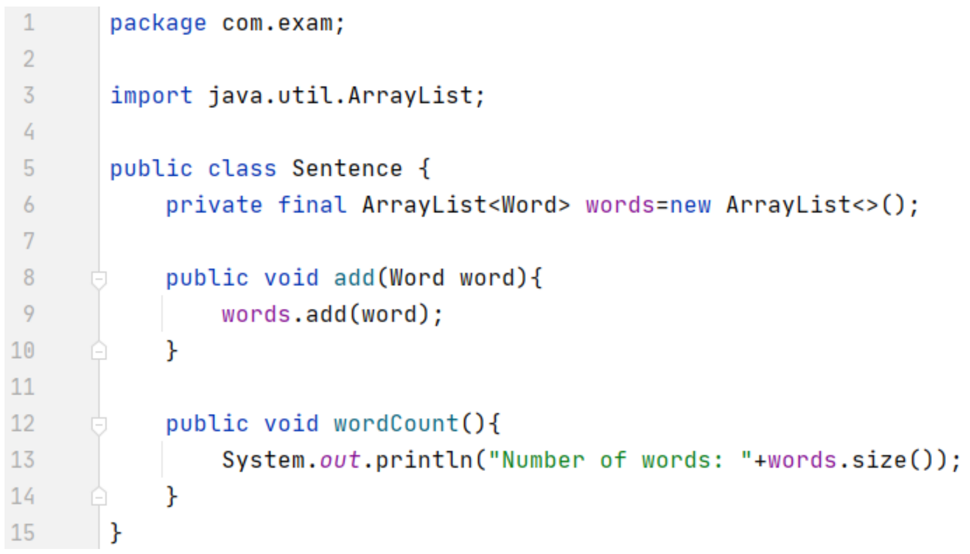
Question 2. Software Maintaince: Which of the following task(s) fall under software maintenance? (more than one answer is possible)

[overall 2 marks]

a. [✓ Fixing existing coding errors]
b. Developing a user survey
c. [✓ Extending a class by adding a new method]
d. Increasing the user base (number of people using the software)

Question 3. Object-Oriented Design: Given the code snippet below, which type of relationship is implemented?

[overall 3 marks]

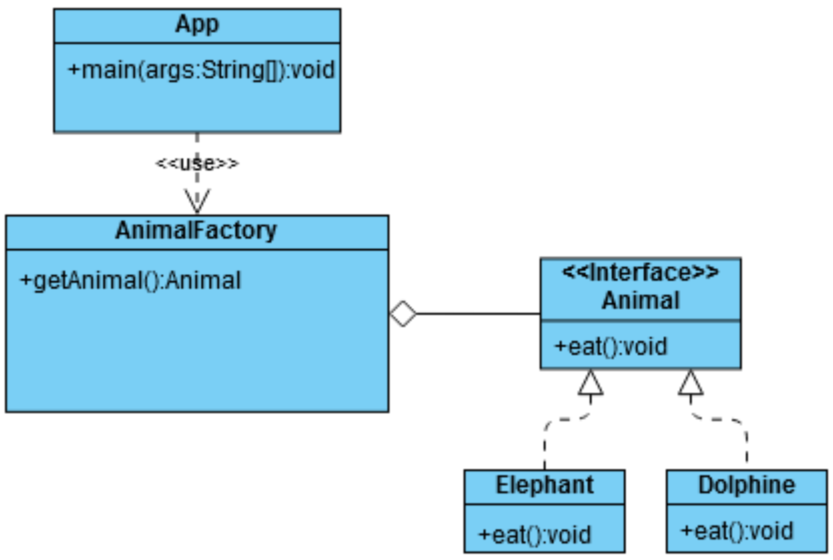


```java
package com.exam;

import java.util.ArrayList;

public class Sentence {
    private final ArrayList<Word> words=new ArrayList<>();

    public void add(Word word){
        words.add(word);
    }

    public void wordCount(){
        System.out.println("Number of words: "+words.size());
    }
}
```

a. Dependency
b. Aggregation
c. Composition
d. Association
e. No Relationship
f. All a-d relationships

Question 4. Design Patterns: The diagram on the next page shows a reverse engineered UML class diagram for creating different types of animals.

[overall 12 marks]



a. What design pattern is it using? [3 marks]
b. When should this pattern be used and why? [5 marks]
c. Provide another brief practical usage scenario (different from that represented above) and justify why the pattern helps [4 marks].

Question 5. Object-Oriented Design:

a. Explain the "Single Responsibility Principle" in your own words. [5 marks]
b. Provide an example related to the **ZooProject** to show how it could be used. [5 marks]

[overall 10 marks]

Question 6. Version Control: You are trying to maintain some source code which is controlled using a git repository. You make some changes locally, but when you try to push to the server git will not let you as someone has modified the branch you are pushing to. Put the steps below in the correct order to merge the local changes into the remote repository.

[overall 8 marks]

2 a. Manually edit the local files to fix the conflict and save the files
1 b. Pull down changes from the remote repository
4 c. Commit the local changes with a suitable commit message
3 d. Add the changed local files
5 e. Push the changes to the remote repository

Question 7. Legacy Code: You have been asked to improve a codebase which is two years old. Which ONE of these sets of steps would you choose to improve this legacy code?

[overall 3 marks]

a) 1. Draw a class diagram of the system.
   2. Write the minimum necessary classes from the diagram.
   3. Write the tests.
   4. Write just enough code to allow the tests to pass.
   5. Refactor to simplify and neaten the code, remove duplication etc.

b) 1. Pull the legacy code and make a repository
   2. Swap the dependencies.
   3. Refactor and debug code.
   4. Write tests to cover the legacy code.
   5. Create fake/dummy objects for components such as databases.
   6. Push the maintained code to production.

c) 1. Write the tests.
   2. Refactor to simplify and neaten the code, remove duplication etc.
   3. Draw a class diagram of the system.
   4. Write the minimum necessary classes from the diagram.
   5. Write just enough code to allow the tests to pass.

[✓ d)] 1. Pull the legacy code and make a local repository.
   2. Write test to cover the legacy code.
   3. Create fake/dummy objects for components such as databases
   4. Swap the dependencies.
   5. Refactor and debug.
   6. Push the maintained code to production.

Question 8. Build Files: Build scripts are sets of instructions for how a project should be compiled, tested, and deployed. Setting up build scripts is helpful for software projects to help improve maintainability. List TWO examples of build tools [2 marks], and describe two differences between how those particular build tools work. [2 marks per difference]

[overall 6 marks]

Question 9. Debugging Source Code: Which of the following strategies could a developer use to debug source code? Answer all those that apply.

[overall 6 marks]

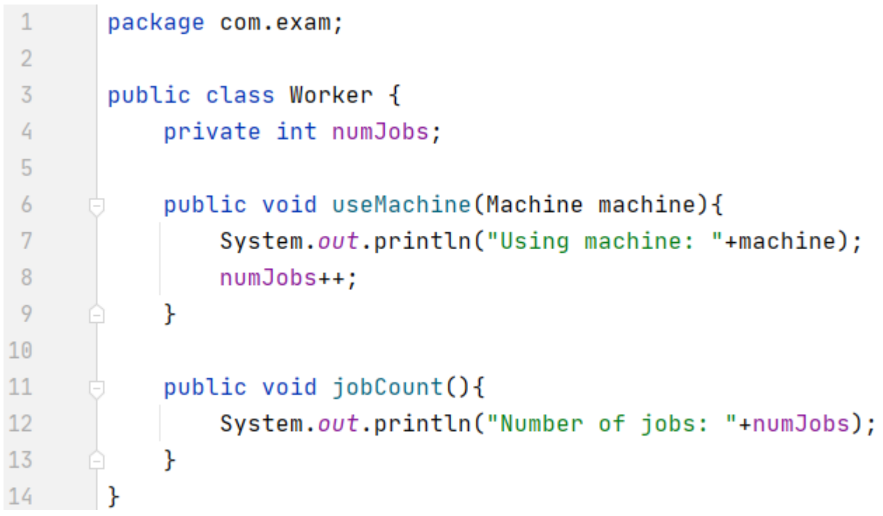
a. Talk to other developers with relevant knowledge.
b. Follow Coding Conventions.
c. Follow code comment conventions.
[✓ d.] Extract working build files.
e. Use test and m_Trace flags
f. Use accessor methods.
g. Review terminating conditions of FOR loops

Question 10. Object-Oriented Design: You are in a team of developers who have just started to work on altering the **ZooProject**. Luckily, the software has been written using good object-oriented coding practice and the original programmers have used a lot of encapsulation in the system. Provide a code example in relation to the **ZooProject** that would demonstrate encapsulation [4 marks]. Then provide 2 reasons with short explanations why using encapsulation in software development is a helpful approach [3 marks each]

[overall 10 marks]

Question 11. Implementing UML: Given the code snippet below, which type of relationship is implemented?

[overall 3 marks]


```java
package com.exam;

public class Worker {
    private int numJobs;

    public void useMachine(Machine machine){
        System.out.println("Using machine: "+machine);
        numJobs++;
    }

    public void jobCount(){
        System.out.println("Number of jobs: "+numJobs);
    }
}
```

[✓ a. Dependency]
b. Aggregation
c. Composition
d. Association
e. No relationship
f. All relationships a-d

Question 12. GUI Development: The MVC pattern consists of model, view, and controller components. Which part of the pattern would you need to investigate, if you were asked to change the access to a backend database?

model

[overall 3 marks]

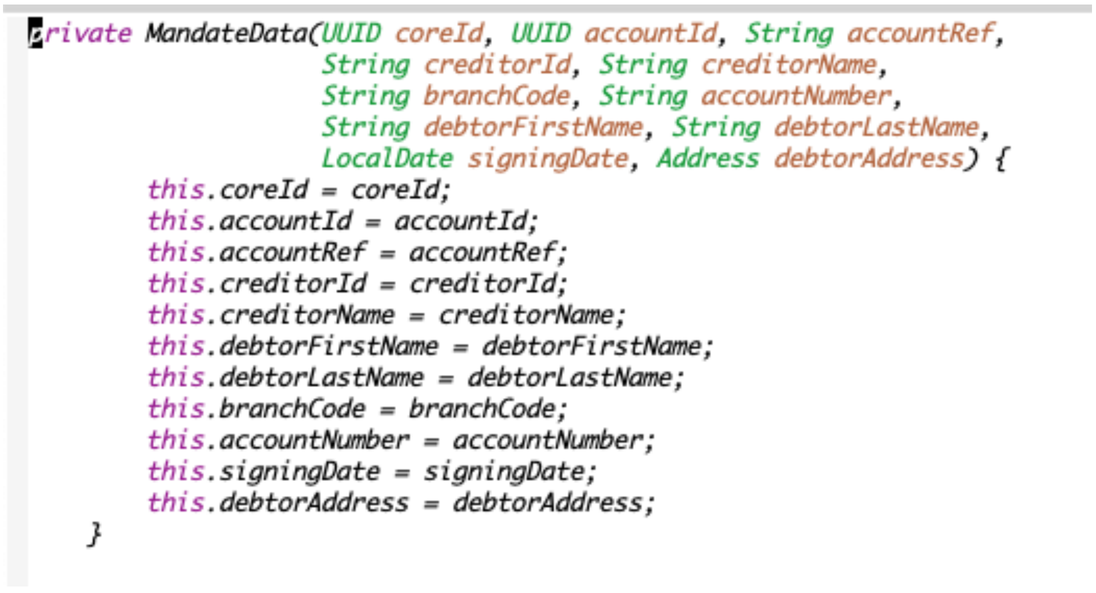
Question 13. Coding Conventions: Bob's Concise Coding Conventions are designed to improve:__________.

maintainability

[overall 2 marks]

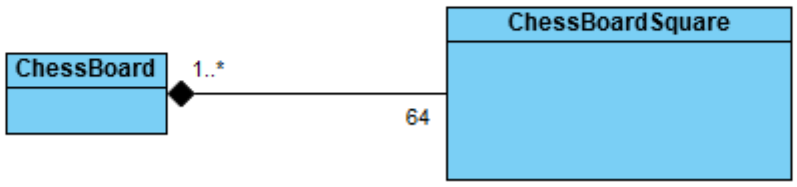
Question 14. Coding Conventions: Given the source code example below. Which of Bob's Coding Conventions are violated? You may refer to individual rules with one or two words.

[overall 3 marks]



```
private MandateData(UUID coreId, UUID accountId, String accountRef,
                    String creditorId, String creditorName,
                    String branchCode, String accountNumber,
                    String debtorFirstName, String debtorLastName,
                    LocalDate signingDate, Address debtorAddress) {
        this.coreId = coreId;
        this.accountId = accountId;
        this.accountRef = accountRef;
        this.creditorId = creditorId;
        this.creditorName = creditorName;
        this.debtorFirstName = debtorFirstName;
        this.debtorLastName = debtorLastName;
        this.branchCode = branchCode;
        this.accountNumber = accountNumber;
        this.signingDate = signingDate;
        this.debtorAddress = debtorAddress;
    }
```

Question 15. UML: A chess board consists of 64 chessboard squares. Given the UML diagram below, answer the following:



[overall 5 marks]

composition

a. What kind of relationship is presented here? [2 marks]
b. There is a mistake in the drawing. What is wrong and why? [3 marks]

End