The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, AUTUMN SEMESTER 2021-2022

Developing Maintainable Software

Time allowed: 2 Hours 0 Minutes

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced

Answer ALL questions

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

ADDITIONAL MATERIAL: Any illustrations may be done by hand or by software such as powerpoint.

INFORMATION FOR INVIGILATORS: none

Note: You can answer the questions in any order. Have a quick look through the exam before you start.

Question 1. Identifying Objects: Given the partial specification below, provide a list of candidate objects. [1 mark per object]

[overall 10 marks]

Automated Teller Machine Partial Requirements Specification

"An automated teller machine (ATM) is a machine through which bank customers can perform a number of the most common financial transactions. The machine consists of a display screen, a bank card reader, numeric and special input keys, a money dispenser slot, a deposit slot, and a receipt printer. When the machine is idle, a greeting message is displayed. The keys and deposit slot will remain inactive until a bank card has been entered."

Question 2. Object-Oriented Design and Protocols: Focussing on one aspect of the partial specification of the ATM *"When the machine is idle, a greeting message is displayed"*, provide a list of candidate protocols for this responsibility. [2 marks per complete protocol]

[overall 10 marks]

Question 3. Software Maintenance: 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 a user base

AC

Question 4. 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 5. 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?

d

[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 6. 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]

Maven Gradle

[overall 6 marks]

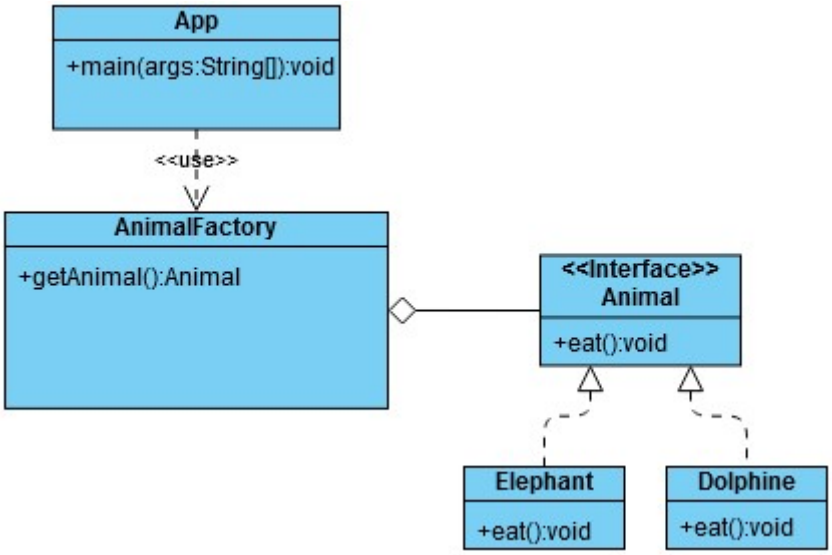
Question 7. Debugging Source Code: Which of the following strategies could a developer use to debug source code? Answer all those that apply.

[overall 6 marks]

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

Question 8. Design Patterns: The diagram below 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 9. 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 GUI?

[overall 3 marks]

controller

Question 10. 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;
    }
```

class variables name
method naming
method parameters

Question 11. Design Patterns: event. execution of a program

a. Briefly explain what an exception is in Java. [2 marks]
b. Provide an example that relates to the partial ATM requirements specification. [3 marks]
c. Briefly explain what a software design pattern is. [3 marks]
d. Provide an example of a design pattern that relates to the partial ATM requirements specification. You are encouraged to use an illustration in your answer. [3 marks]
e. What do exceptions in Java and software design patterns have in common? [4 marks]

[overall 15 marks]
[Allowed maximum length of your answer: 200 words]

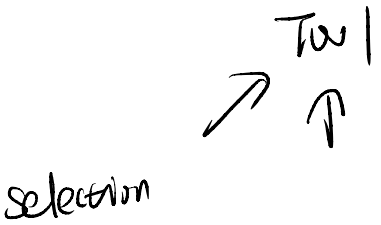
anti-pattern

Question 12. Design Patterns: Provide two reasons why it is advantageous to encapsulate data access in the MVC design pattern. [2 marks each]

[overall 4 marks]
[Allowed maximum length of your answer: 100 words]

Question 13. Object-Oriented Design: Given the following list of classes, arrange them in an object-oriented class hierarchy: 1) Tool, 2) Selection Tool, 3) Creation Tool, 4) Ellipse Tool, 5) Line Tool, 6) Rectangle Tool, 7) Text Tool. Show your design using an illustration.

[overall 7 marks]

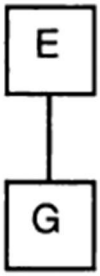


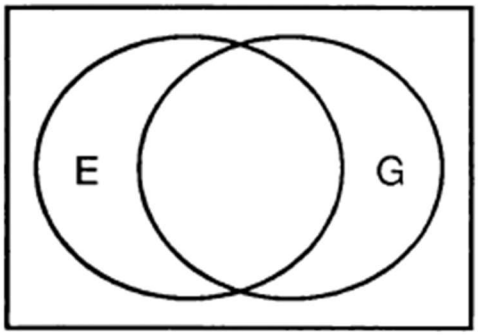
Question 14. Object-Oriented Design: Which of the following kinds of relationships are valid in object-oriented design. Identify/List each valid type of relationship:

1. Is-Copy-Of
2. Is-Part-Of
3. Is-Kind-Of
4. Is-Relative-Of
5. Is-Associated-With
6. Is-Type-Of

[overall 4 marks]

Question 15. Object-Oriented Design: The illustrated class hierarchy below is incorrect because the subclass (child) shown only supports part of the responsibilities of the superclass (parent). Can you revise this class hierarchy in order to fix this problem? Illustrate the corrected class hierarchy using a figure or illustration.





[overall 7 marks]

End