The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 4 MODULE, SPRING SEMESTER 2011-2012

OBJECT ORIENTED SYSTEMS (G64OOS)

Time allowed TWO hours

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

Section A: Answer Question 1
AND
Section B: Answer TWO of the remaining four questions

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 your examination paper over until instructed to do so

ADDITIONAL MATERIAL:

- Exam paper

INFORMATION FOR INVIGILATORS:

- ***PLEASE ENSURE THAT EXAM PAPER AND ANSWER BOOK ARE BOTH SUBMITTED***

SECTION A

Question 1: OO Analysis and Design [overall 50 marks]

(a) Name and briefly describe the purpose of four of the 26 different UML diagram types. State for each whether they are structural or behavioural.

(12 marks)

Assuming you are describing the process of filling up a car at a self-service petrol station where you pay by credit-card at the pump:

(b) Provide the use case diagram plus two other Object Oriented Analysis/Design UML diagrams of your choice (make sure your diagrams are consistent)

(30 marks)

(c) Provide the highest-level use case (including pre- and post-conditions and alternative paths)

(8 marks)

SECTION B

Question 2: OO Design Principles [overall 25 marks]

Fig. 1describes the first draft of a class diagram describing the process of baking a cake. Please add your answers for Q2.b – Q2.e directly into Fig.1 and ***SUBMIT THE EXAM PAPER TOGETHER WITH THE ANSWER BOOK AT THE END OF THE EXAM.***

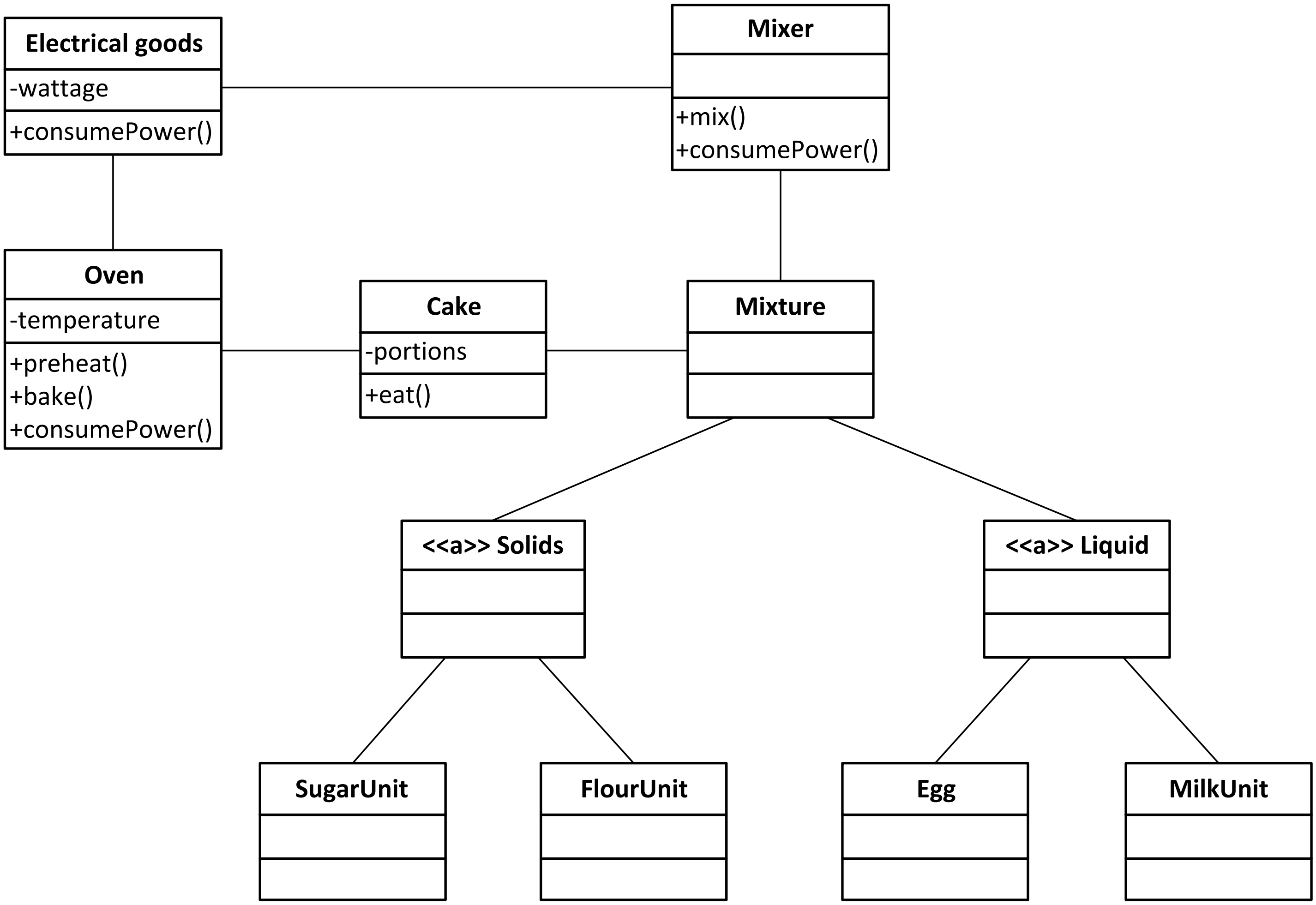


Figure 1: Draft of class diagram describing the process of baking a cake

(a) List and briefly explain at least four different OO principles that appear in the class diagram.

(8 marks)

(b) Circle and label in the class diagram (directly in Fig. 1) the OO principles you provided as the answer to Q2.a (at least one occurrence per OO principle). Write down a brief justification for each decision.

(4 marks)

(c) Mark in the class diagram (directly in Fig. 1) where necessary whether a relation is a derivation, aggregation, or composition to the class diagram. Your diagram should have at least one of every relationship type.

(5 marks)

(d) Add the multiplicity of relations where appropriate (directly in Fig. 1).

(3 marks)

(e) Extend the graph by adding a differentiation of cake types (directly in Fig. 1).

(5 marks)

Question 3: OO Design Principles [overall 25 marks]

(a) Describe four out of the five SOLID principles and give an example how each is applied.

(10 marks)

(b) In what way does the following example shown in Fig. 2 violate the SOLID principles?

(10 marks)

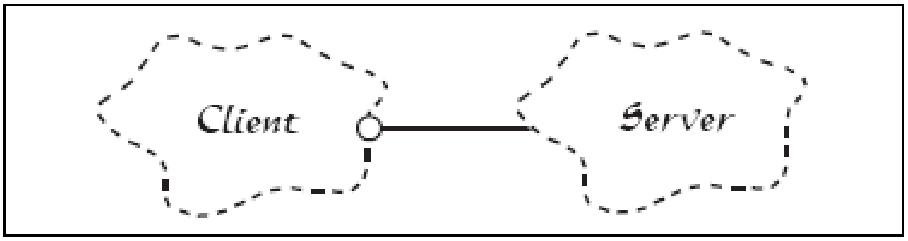


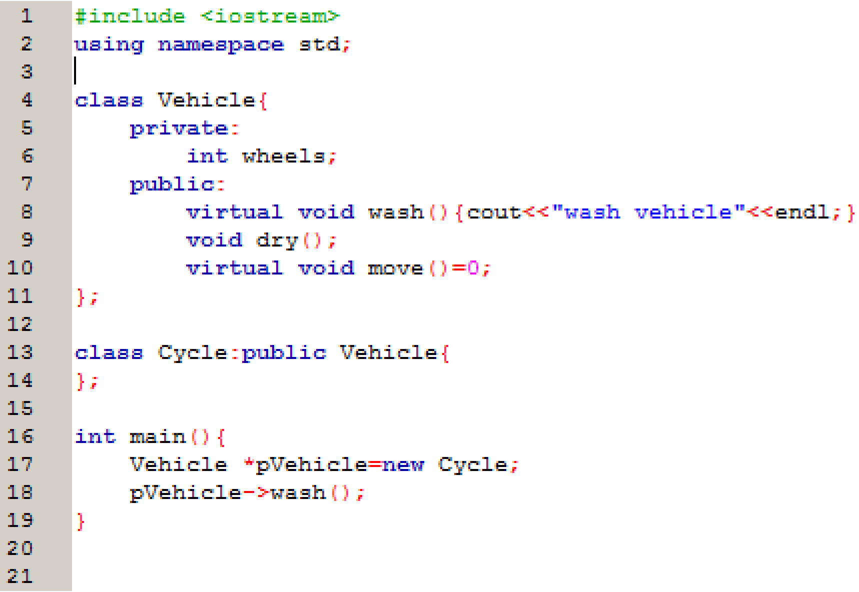
Figure 2: A client-server class diagram violating the SOLID principles

(c) Draw a class diagram that would be a suitable extension of the diagram given above so that the new diagram adheres to the SOLID principles. Briefly justify your decision.

(5 marks)

Question 4: OO Programming (inheritance and polymorphism) [overall 25 marks]

Have a look at the provided source code in Fig. 3.



```
1   #include <iostream>
2   using namespace std;
3   
4   class Vehicle{
5       private:
6           int wheels;
7       public:
8           virtual void wash(){cout<<"wash vehicle"<<endl;}
9           void dry();
10          virtual void move()=0;
11  };
12  
13  class Cycle:public Vehicle{
14  };
15  
16  int main(){
17      Vehicle *pVehicle=new Cycle;
18      pVehicle->wash();
19  }
20  
21  
```

Figure 3: Erroneous piece of code

(a) Which line in the code is causing a problem and why?

(7 marks)

(b) Add the minimum amount of code to solve this problem (points will be deduced if you provide unnecessary code)!

(5 marks)

(c) When would you use abstract classes? How do you make a concrete class abstract?

(5 marks)

(d) Explain the difference between overriding and overloading.

(4 marks)

(e) Can overriding and overloading both achieve polymorphism? Briefly explain your answer.

(4 marks)

Question 5: Agile Software Methodology [overall 25 marks]

(a) What are the key differences between the ***agile***, ***waterfall***, and ***spiral*** software development processes?

(6 marks)

(b) Describe and contrast the following two types of agile software methodology: Scrum and Extreme Programming

(6 marks)

(c) Briefly describe the test-driven development cycle. How does it link with the agile methods Scrum and Extreme Programming?

(6 marks)

(d) Your customer wants to create a website on which they can store, organise, and search through all the image and video data they have collected over their lifetime. Create a time planning for this project using the Scrum methodology and describe how the time planning is updated while the project progresses.

(7 marks)

End