13/5/2019

09.30 - 11.30am

CMPU 2020 Software Engineering 2

KE 102, KE 104, KE 154, Kevin Street

Programme Code: DT228
Module Code: CMPU 2020
CRN: 22394

TECHNOLOGICAL UNIVERSITY DUBLIN

KEVIN STREET CAMPUS

BSc. (Honours) Degree in Computer Science

Year 2

SEMESTER 2 EXAMINATIONS 2018/19

Software Engineering 2

Mr. Richard Lawlor
Dr. Deirdre Lillis
Dr. Martin Crane

Instructions

Attempt four out of five questions
All questions carry equal marks

Two hour exam

1. **(a)** Comment on the differences between unit testing and integration testing.

(15 marks)

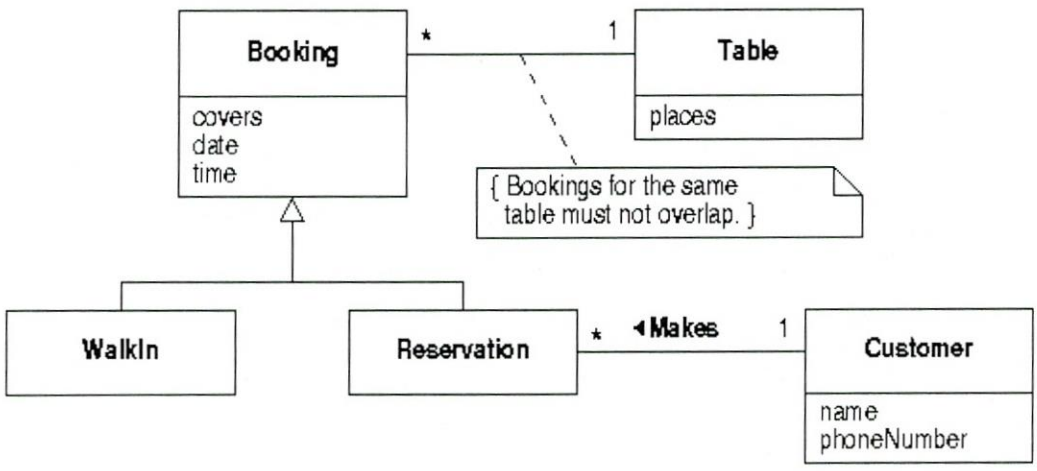
(b) Given the following code unit, write a JUnit test class which sets up a single test on the **evaluate()** operation.

```
public class Calculator {
  public int evaluate(String expression) {
    int sum = 0;
    for (String summand: expression.split("\\+"))
      sum += Integer.valueOf(summand);
    return sum;
  }
}
```

(10 marks)

2 **(a)** Given the following initial domain model of a restaurant booking system, show how it could be refined by performing use-case realisations for the use-cases "*Display Booking*" and "*Record an Arrival*".

Briefly comment on your refinements at each stage.



(15 marks)

(b) In light of the use-case realisation from part (a), provide a more comprehensive analysis level class diagram which includes the above class diagram, showing the control class and any other appropriate classes.

(10 marks)

3. You are required to do some object-oriented design for a standalone restaurant software system that mainly manages bookings. The restaurant software should be able to handle advance reservations, walk-in bookings, assigning tables to reservations and so on.

(a) A layered architecture allows for separation of concerns. Explain what is meant by this. Then describe an appropriate layered architecture for the restaurant system given that it will be implemented as standalone software.

(5 marks)

(b) In light of your architecture from part (a), suggest an appropriate design which would allow for persistency/storage concerns without compromising the cohesion of the application classes. Comment on the reasons for your design choices.

(10 marks)

(c) In the mapper Java classes a Hashtable may be used as in:

```
public class CustomerMapper
{
    private Hashtable cache ;
    ...
    ...
}
```

For this class, outline the steps involved when the operation **getCustomer(String n, String p)** is called and in particular how it interacts with the Hashtable and why. Code is not required.

Mention any design pattern implicit in this class and provide the corresponding code fragment.

(10 marks)

4. (a) Explain what is meant by *Design by Contract* (DbC). Elaborate on how a contract is affected by subclassing/polymorphism.

(9 marks)

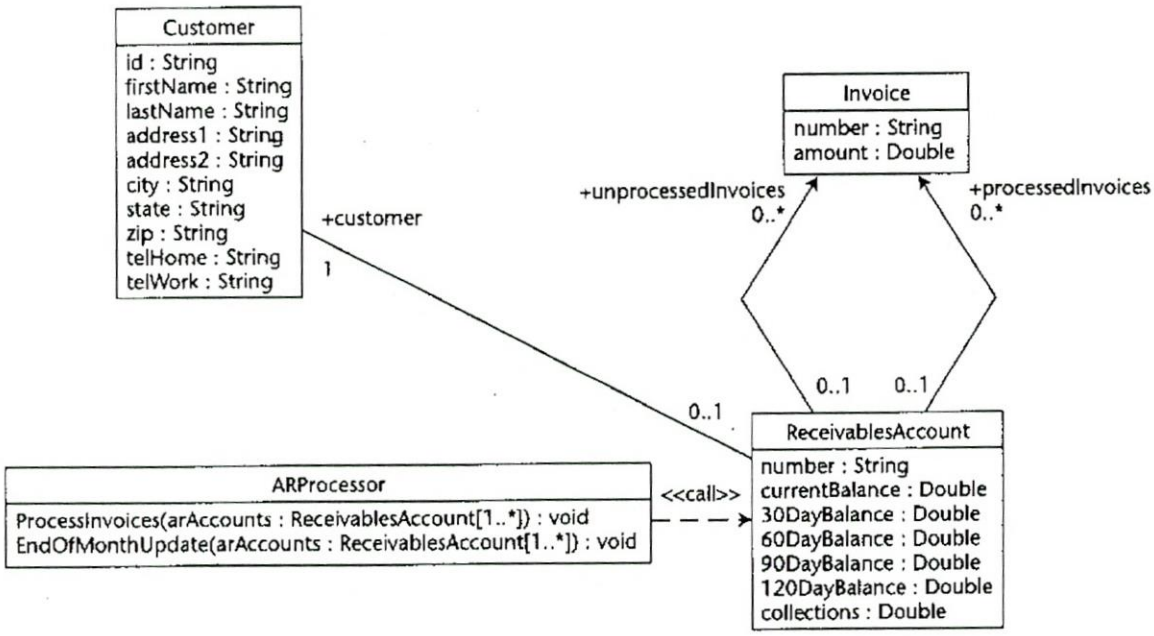
(b) Within the context of DbC, comment on benefits and obligations for both client code and provider code. Mention when exceptions might be appropriate.

(6 marks)

(c) Given the class diagram below, write an *Object Constraint Language* (OCL) contract invariant for *ReceivablesAccount* which states that no invoice can be in both *processedInvoices* and *unprocessedInvoices* collections at the same time.

Write an OCL contract that you deem appropriate to express the business logic *ProcessInvoices()* operation of the class *ARProcessor*.

(10 marks)



5. (a) Describe six of the key practices of the agile methodology XP.

(12 marks)

(b) Discuss the diagram below from the point of view: Anticipatory Design versus Refactoring.

(13 marks)

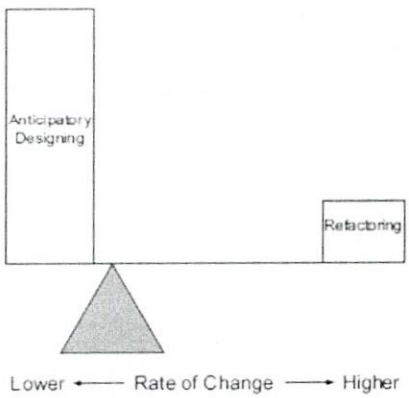


Figure 3 — Balancing design and refactoring, pre-internet.

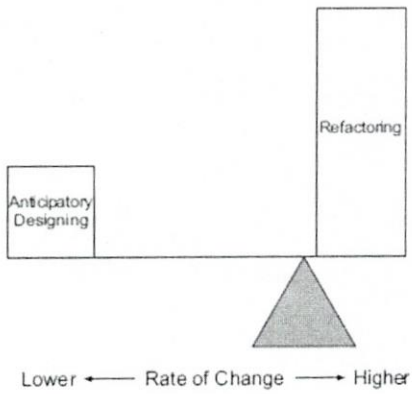


Figure 4 — Balancing design and refactoring today.