BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

OBJECT ORIENTED PROGRAMMING

Tuesday 21st March 2017 - Afternoon
Answer **any** FOUR questions out of SIX. All questions carry equal marks
Time: TWO hours

Answer any Section A questions you attempt in Answer Book A
Answer any Section B questions you attempt in Answer Book B

The marks given in brackets are **indicative** of the weight given to each part of the question.

Calculators are **NOT** allowed in this examination.

SECTION A
Answer Section A questions in Answer Book A

A1.

a) Explain what is meant by the following terms:

(i) Structured programming;
(ii) Procedural programming;
(iii) Abstract data types;
(iv) Typed language;
(v) Untyped languages;

(15 marks)

b) Define the terms *coupling* and *cohesion* in the context of object oriented programming. Within your discussion, explain how *coupling* and *cohesion* can lead to either good or bad software design.

(10 marks)

A2.

a) Explain what is meant by a *design pattern* in the context of object oriented programming.

(5 marks)

b) Describe in detail TWO *design patterns* that you are familiar with, stating the motivation for the pattern, including a UML class diagram for the pattern and an explanation of the classes which participate in the pattern.

(20 marks)

A3.

a) Define what the following Object Constraint Language (OCL) concepts mean:

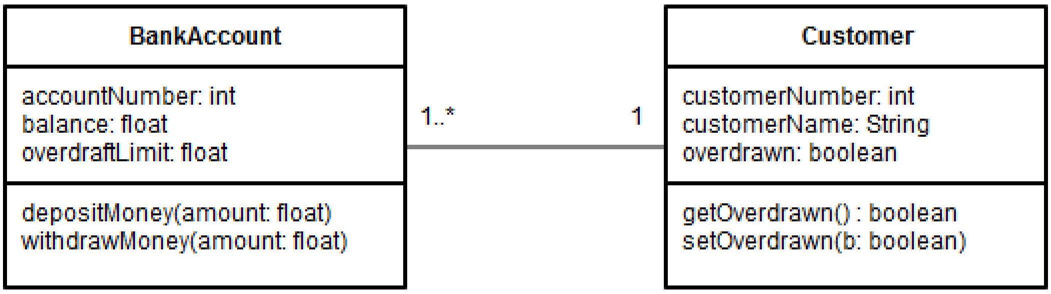
(i) Invariant;
(ii) Pre-condition;
(iii) Post-condition.

(9 marks)

b) Explain the role of the OCL in the development of an object oriented system.

(10 marks)

c) Given the following UML class diagram:



Explain what the following OCL statement: means:

```
context: BankAccount ::withdrawMoney(amount: float)
pre:     amount > 0 and Customer.getOverdrawn() = false
post:    balance = balance@pre - amount
         if amount > balance@pre + overdraftLimit
         then Customer.setOverdrawn(true)
         endif
```

(6 marks)

SECTION B
Answer Section B questions in Answer Book B

B4.

The atoms of different elements have different numbers of protons, neutrons and electrons. Electrons are negatively charged, protons are positively charged, and neutrons have no charge.

a) In an object oriented programming language of your choice, write a definition for an `atom` class that contains:

(i) fields for storing the numbers of `protons`, `neutrons` and `electrons` with appropriate visibility;

(ii) setter and getter methods for manipulating these fields, ensuring that the minimum value for `electrons` and `protons` is 1, and the minimum value for `neutrons` is 0;

(iii) a constructor that initialises new objects of `atom` to be the smallest element (Hydrogen), for which the number of `protons` is 1, the number of `neutrons` is 0, and the number of `electrons` is 1.

(15 marks)

b) Write a new method for the `atom` class called `isIon` that will return true or false, depending upon whether the atom is an ion. An atom is an ion if it is charged (i.e., if the number of electrons ≠ the number of protons).

(5 marks)

c) Write a new method for the `atom` class called `getAtomicMassNumber` that will calculate and return the atomic mass number of the atom. Atomic mass number of an atom (often denoted A) is defined as the number of protons plus the number of neutrons.

(5 marks)

B5.

Consider the code fragment written below:

```
public class A
{
  private int a;
  protected int b;
  public int c;
  public A();
  public void seta(int new_a);
  public void setb(int new_b);
  public void setc(int new_c);
  public int geta();
  public int getb();
  public int getc();
};

public class B extends A
{
  private int d;
  public B();
  public void setd(int new_d);
  public int getd();
}
```

a) State the relationship between class `A` and class `B`, and show how this code fragment would be represented in a UML class diagram.

(10 marks)

b) State the name of one other kind of inter-class relationship, and show both a code fragment in which this relationship is implemented, and how it would be represented in a UML class diagram.

(15 marks)

B6.

a) Define *compile time polymorphism*, and provide a code fragment that implements this concept in an object oriented programming language of your choice.

(10 marks)

b) Define *run-time polymorphism*, and provide a code fragment that implements this concept in an object oriented programming language of your choice.

(15 marks)