

**BCS The Chartered Institute for IT**  
**THE BCS PROFESSIONAL EXAMINATIONS**  
**BCS Level 5 Diploma in IT**  
**OBJECT ORIENTED PROGRAMMING**

Wednesday 28<sup>th</sup> April 2010 – Morning

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 <b>NOT</b> allowed in this examination.
---

**Section A**

Answer Section A questions in Answer Book A

A1.

- a) How do *mock objects* and *fake objects* differ?  

**(5 marks)**
- b) How do *ad-hoc polymorphism* and *parametric polymorphism* differ?  

**(5 marks)**
- c) Outline the problems associated with the use of *fragile base classes*.  

**(5 marks)**
- d) Outline the principle of *substitutability*.  

**(5 marks)**
- e) When is it appropriate to use *virtual inheritance*?  

**(5 marks)**

A2.

- a) Distinguish between *association*, *aggregation* and *composition*, indicating how these are represented in a UML class diagram.

**(10 marks)**

- b) Distinguish between *abstract classes* and *concrete classes*, indicating how these are represented in a UML class diagram.

**(5 marks)**

- c) How are *private*, *protected*, *public* and *package* members identified in a UML class diagram?

**(5 marks)**

- d) How does an *object diagram* differ from a *class diagram* in the UML?

**(5 marks)**

A3.

- a) Describe three practical examples of *polymorphism*. Provide code examples to illustrate your answer.

**(15 marks)**

- b) Distinguish between the following pairs of methods.

- i) *abstract* and *concrete*
- ii) *accessor* and *mutator*
- iii) *support* and *service*
- iv) *constructor* and *destructor*
- v) *instance* and *class*

**(10 marks)**

### **Section B**

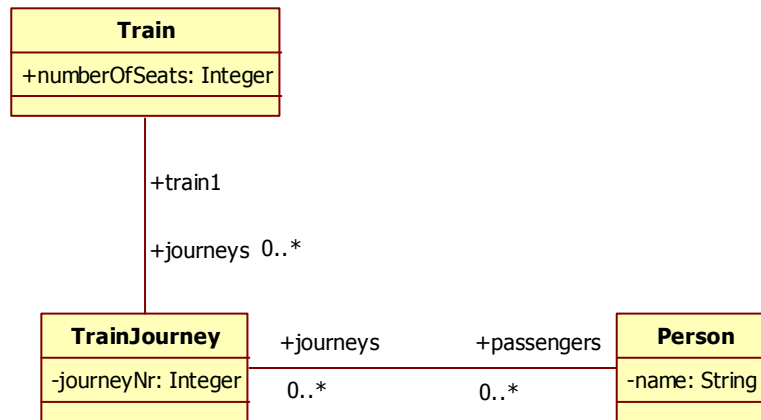
Answer Section B questions in Answer Book B

B4.

- a) Write a report which explains to a software team manager how the UML can contribute to the development of object oriented software.

**(10 marks)**

- b) Describe the meaning captured by the following class diagram and OCL statement



**context** Train Journey

**inv:** passengers->size() <= train.number Of Seats

**(15 marks)**

B5.

- a) Describe two testing techniques which may be used to ensure that methods produce the expected results.

**(10 marks)**

- b) Object oriented systems almost invariably involve a number of interacting objects. Describe an approach which can be used to test the interactions

**(10 marks)**

- c) Discuss the following statement:

“Program testing can be used to show the presence of bugs, but never to show their absence!” (Dijkstra)

**(5 marks)**

B6.

a) Explain the following terms:

- i) *Structured programming*;
- ii) *Modular programming*;
- iii) *Abstract data types*;
- iv) *Typed languages*;
- v) *Untyped languages*.

**(10 marks)**

b) Define the terms *coupling* and *cohesion* and explain how these concepts contribute to the quality of a program. Show how the object oriented concept of *encapsulation* aids a programmer to produce good quality code when measures of coupling and cohesion are used to gauge quality.

**(10 marks)**

c) Contrast the way in which classes are implemented in typed object oriented programming languages and untyped object oriented programming languages.

**(5 marks)**