

BCS THE CHARTERED INSTITUTE FOR IT
BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

OBJECT ORIENTED PROGRAMMING

Tuesday 24th March 2015 – 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) Define the following terms:

- i) Abstract data type;
- ii) Collection class;
- iii) Class library.

(6 marks)

b) A queue is a first in, first out linear data structure. A queue can have any object as an element. It is characterised by two fundamental operations, called *enqueue* and *dequeue*. The *enqueue* operation adds a new item to the back of the queue. If the space allocated to hold the queue is full when the *enqueue* operation is attempted then an error condition is raised. The *dequeue* operation removes an item from the front of the queue. A *dequeue* moves previously added items towards the front of the queue, or results in an empty queue. If the queue is empty when a *dequeue* operation is attempted then an error condition is raised.

Using an object oriented programming language with which you are familiar, write code which implements a queue. Your code should store the queue elements in an array and should not make use of a queue class from a class library. **(19 marks)**

A2.

a) Explain what is meant by a design pattern.

(5 marks)

b) Describe five characteristics you could use to document a design pattern.

(10 marks)

c) Give an example of code which uses the iterator pattern.

(10 marks)

A3.

a) Explain the following terms:

- i) Class;
- ii) Object;
- iii) Class variable;
- iv) Method overloading;
- v) Constructor.

(10 marks)

b) Explain why it is sometimes useful to overload a constructor.

(5 marks)

c) Using an object oriented programming language with which you are familiar write a class which contains a method called `getNumberOfInstances`. This method should return the number of instances of the class.

(10 marks)

Section B**Answer Section B questions in Answer Book B**

B4. Consider the following class definition that represents an AM radio:

```
public class radio
{
    public bool isSwitchedOn;    // either true or false
    public int volume;          // from 0 to 10
    public double frequency;    // from 535 to 1605 Mhz
};
```

a) Provide a redesigned **radio** class that uses more appropriate member access operators.

(5 marks)

b) Provide signatures for two mutator (setter) methods that will enable the two instance variables to be changed through an object of the class.

(5 marks)

c) Write bodies for the mutator (setter) methods for which you provided signatures in question (b), above. The implemented methods must prevent the two instance variables from being set to invalid values. Return a boolean flag to indicate the success/failure of the requested operation.

(5 marks)

d) Provide a constructor for the **radio** class that will initialise the instance variables to suitable (valid) start values.

(5 marks)

e) Write a short test harness that instantiates the **radio** class and demonstrates that the setter methods you designed above behave correctly, using a set of boundary tests.

(5 marks)

B5. The follow questions relate to class diagrams represented in the Unified Modelling Language (UML).

a) State the meaning of the following member visibility operators:

- (i) #
- (ii) -
- (iii) $\bar{}$
- (iv) /
- (v) +

(5 marks)

b) Draw or state how the following inter-class relationships are represented:

- (i) composition
- (ii) aggregation
- (iii) inheritance
- (iv) dependency
- (v) realization.

(5 marks)

c) State the meaning of the following inter-class relationship multiplicity indicators:

- (i) 1..*
- (ii) *
- (iii) 0..1
- (iv) 2
- (v) 0..*

(5 marks)

d) Show how abstract and concrete classes are represented in a UML class diagram, and briefly describe an example use of these class types. **(10 marks)**

B6.

a) Distinguish between structural and behavioural UML diagrams, briefly describing one example of each (note: you may not use class diagrams as your example). **(5 marks)**

b) Distinguish between multiple and multi-level inheritance. **(5 marks)**

c) Distinguish between static and dynamic binding. **(5 marks)**

d) Distinguish between ad-hoc and parametric polymorphism. **(5 marks)**

e) Distinguish between virtual methods and pure virtual methods. **(5 marks)**

