

BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

OBJECT ORIENTED PROGRAMMING

Friday 4th October 2024 – 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) Object-oriented languages provide **three** basic member access modifiers.

For **each** modifier, explain their purpose and describe example scenarios to illustrate your answer. Your scenarios should include both data members and functions.

(15 marks)

- b) Explain **two** different ways that object-oriented programming languages promote code re-use that are **not** also found in procedural languages.

Illustrate your answer with examples.

(10 marks)

A2.

- a) Write a class definition for a countdown timer in an object-oriented language that you are familiar with.

It should include:

- data members to store the time (hours, minutes and seconds).
- setter/getter functions to adjust and find out how much time is remaining.
- a tick function that decreases the timer by 1 second. Once the timer reaches 0, the message “alarm” is to be displayed.

Use appropriately named identifiers throughout.

(20 marks)

- b) Write a corresponding main function in which you give an example object of your class, set the timer duration, and enter a loop that invokes the tick function until the timer reaches 0.

(5 marks)

A3.

- a) Explain what is meant by polymorphism in object-oriented programming. Give **one** benefit of polymorphism.

(3 marks)

- b) Describe **three** different ways that object-oriented programming languages implement the concept of polymorphism, illustrating your answer with code fragments.

(12 marks)

- c) Compare and contrast abstract and concrete classes making at least **five** distinctions.

You should present your answer in the form of a two-column table.

(10 marks)

Section B
Answer Section B questions in Answer Book B

The Walking Group

The Midlands Ramblers is a walking group that arranges walks for its members. New members can register their details online, by providing their name, address, telephone number, and emergency contact name and telephone number.

At least two walks take place during the week. These activities are advertised in a Group Calendar, which is produced and put online by a Committee Member.

Members can record details of their own walks online. By default, this information is for the use of the member only, but they can elect to make it public to the rest of the group.

During the year, the group runs external events that are part of Walking Festivals. These events are open to anyone. Entrants must register for each event separately, even if they are members of the group. Entrants need to provide their name, telephone number, email address, and an emergency contact number. Before the event, a committee member will email the entrant the final details of the walk. Each event has a maximum number of participants and once the limit is reached, no further entries are allowed.

For each event, several administration tasks have to be carried out by a committee member, such as order refreshments and medals. A record needs to be kept of the date these were ordered and the quantity. For insurance reasons, a member of the committee must register the event with the Ramblers Association.

B4.

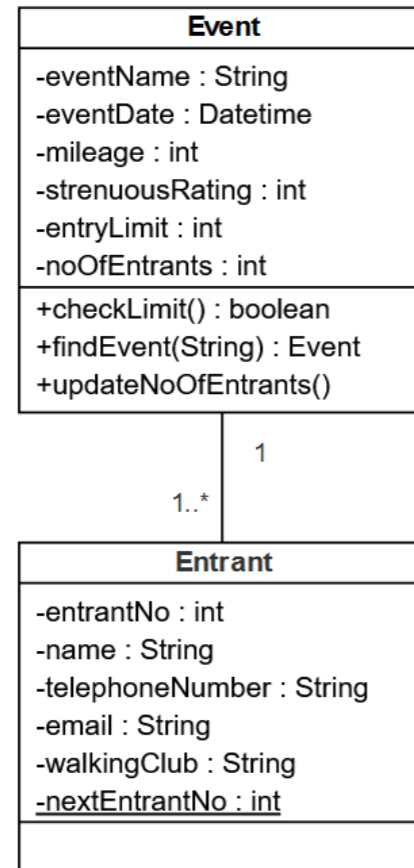
- a) Draw a Use Case diagram for the Midlands Ramblers walking group. **(15 marks)**
- b) Discuss why Use Case diagrams and descriptions are used in the development of an object-oriented system.

You should include an example scenario from the walking group to illustrate points in your answer.

(10 marks)

[Turn Over]

B5.



- a) In an object-oriented programming language that you are familiar with, write code to partially implement the class diagram above, which represents Entrants entering a walking Event.

You **should** write code to implement the following:

- Class definitions with declarations of all attributes (variables).
- A default constructor for each class that sets the variables to appropriate initial values.
- *eventDate* contains the date and time of the event.
- *strenuousRating* shows how strenuous the event is and is an integer between 1 and 5 (default 3).
- *entryLimit* is an integer between 10 and 20 (default 20).
- *updateNoOfEntrants()*, which increments the *noOfEntrants* attribute by one for each entrant and should be called by the Entrant constructor.
- The class variable should be set and incremented appropriately.

You **do not** need to implement the following:

- Setter and getter methods for the attributes.
- Code for the *checkLimit()* method.
- Code for the *findEvent(String)*, but you can assume it exists and returns an instance of the event being entered.

(15 marks)

- b) The *checkLimit()* method above is a method that checks if the number of entrants for a Walking Event has been reached. If so, the method will return true, if not, the method will return false. Write Object Constraint Language (OCL) code for this method.

(5 marks)

- c) Explain how the OCL code could be used for constructing class tests to show that the code is working correctly.

(5 marks)

B6.

- a) Discuss why you should produce the following UML diagrams in the development of an object-oriented system. Include brief examples of their use:

- i. Class diagrams.
- ii. Object interaction diagram.
- iii. Object state transition diagrams.

(15 marks)

- b) Explain what the terms coupling and cohesion mean in the context of software engineering. You should include a clear explanation of what is considered good practice in terms of coupling and cohesion, with justification.

(10 marks)

END OF EXAMINATION