

BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 6 Professional Graduate Diploma in IT

Programming Paradigms

Thursday 28th March - Morning

Answer any THREE questions out of FIVE. All questions carry equal marks.
Time: THREE 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. This question is about imperative and declarative programming and language standardisation.

a) Illustrate the essential features and concepts for both of the following programming paradigms. In your answer, identify example languages and provide examples of statements in the languages that illustrate the following concepts:

- imperative.
- declarative.

(12 marks)

b) Discuss the key reasons why it is important to have an international standard for any programming language. Consider the benefits that standardisation can bring and problems it can prevent.

(13 marks)

A2.

a) What are the key features and functions of language compilers and interpreters? Explain any similarities or differences between them.

(10 marks)

b) Discuss the meaning of code debugging and code testing. Consider any similarities and differences between them. What tool support is available within Integrated Development Environments (IDEs) for these techniques?

(15 marks)

A3. A company wants to create a new software system for drawing diagrams. The diagrams can contain shapes and lines. Example shapes are rectangles, ellipses and triangles. There can also be lines that can optionally have arrows at one or both ends. The software will save and load the diagrams to disk.

The company is going to use an Object Oriented (OO) programming language to build the system.

Describe the core concepts, features and applications of the following aspects of the OO programming paradigm and how they would benefit the company. Use examples and diagrams where appropriate.

a) Classes, Constructors and Objects.

(10 marks)

b) Encapsulation, Methods and Attributes.

(10 marks)

c) Inheritance.

(5 marks)

Section B
Answer Section B questions in Answer Book B

B4. A small clothing shop manages its stock list in a spreadsheet. The company would like to create a website to enable online sales. The company needs to move its stock information into a database to allow concurrent stock queries and purchase updates by many customers over the web. The database needs to have concurrency control for multiple users.

- a) If concurrent transactions by multiple users are interleaved without concurrency control, what problems could occur?

(15 marks)

- b) Discuss solutions available to tackle these problems, and the advantages and disadvantages of these solutions.

(10 marks)

B5. This question is about logic programming.

- a) Logic programming languages such as Prolog usually operate under the closed-world assumption. Explain what this assumption is. Explain why this assumption is made and what implications it has for the reasoning in Prolog. What is 'negation-as-failure'? **(13 marks)**

- b) In the following program we use `\+` to represent negation-as-failure (sometimes called **not** in some Prologs).

```
eats(sally, X) :- vegetarian(X).  
  
vegetarian(X) :- food(X), \+ meat(X).  
  
food(X) :- vegetable(X).  
food(X) :- meat(X).  
food(X) :- dairy(X).  
  
vegetable(bean).  
meat(chicken).  
vegetable(mushroom).  
dairy(cheese).
```

- i) What is the result of asking the following query?

```
eats(sally, X).
```

(6 marks)

- ii) If instead, we replace the line defining the vegetarian predicate:

```
vegetarian(X) :- food(X), \+ meat(X).
```

with the following alternative definition:

```
vegetarian(X) :- \+ meat(X), food(X).
```

what would be the result of the following query?

```
eats(sally, X).
```

Explain your answer fully.

(6 marks)

END OF EXAMINATION PAPER