



---

# A-level COMPUTER SCIENCE

## Paper 1

---

Monday 1 June 2020

Morning

Time allowed: 2 hours 30 minutes

### Materials

For this paper you must have:

- a computer
- a printer
- appropriate software
- the Electronic Answer Document
- an electronic version and a hard copy of the Skeleton Program
- an electronic version and a hard copy of the Preliminary Material.

You must **not** use a calculator.

### Instructions

- Type the information required on the front of your Electronic Answer Document.
- Before the start of the examination make sure your **Centre Number**, **Candidate Name** and **Candidate Number** are shown clearly **in the footer** of every page (also at the top of the front cover) of your Electronic Answer Document.
- Enter your answers into the Electronic Answer Document.
- Answer **all** questions.
- Save your work at regular intervals.

### Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.
- No extra time is allowed for printing and collating.
- The question paper is divided into **four** sections.

### Advice

You are advised to allocate time to each section as follows:

**Section A** – 45 minutes; **Section B** – 20 minutes; **Section C** – 15 minutes; **Section D** – 70 minutes.

### At the end of the examination

Tie together all your printed Electronic Answer Document pages and hand them to the Invigilator.

### Warning

It may not be possible to issue a result for this paper if your details are not on every page of your Electronic Answer Document.

**Section A**

You are advised to spend no longer than **45 minutes** on this section.

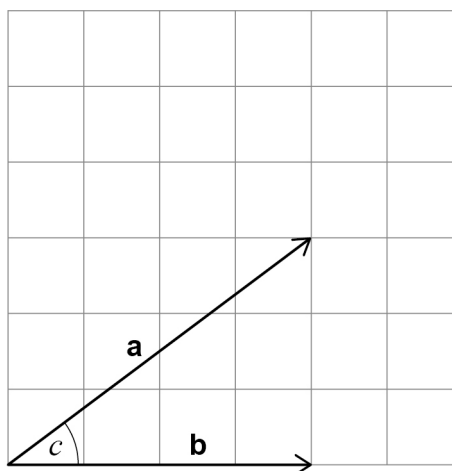
Type your answers to **Section A** in your Electronic Answer Document.

You **must save** this document at regular intervals.

|   |   |
|---|---|
| 0 | 1 |
|---|---|

**Figure 1** shows two vectors **a** and **b** and the angle between them,  $c$ .

**Figure 1**



$$\mathbf{a} = [4, 3]$$

$$\mathbf{b} = [4, 0]$$

The magnitude of a vector, represented as an arrow, is the length of the arrow. The magnitude of vector **a** is 5 because  $\sqrt{4^2 + 3^2} = 5$

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | . | 1 |
|---|---|---|---|

What is the magnitude of vector **b**?

**[1 mark]**

|   |   |   |   |
|---|---|---|---|
| 0 | 1 | . | 2 |
|---|---|---|---|

Calculate the dot product of vectors **a** and **b**. You should show your working.

**[2 marks]**

The angle between two vectors cannot be larger than  $180^\circ$ . If the angle is measured as being greater than  $180^\circ$  then this angle is subtracted from  $360^\circ$  to find the actual angle between the two vectors.

Example: if an angle between two vectors represented as arrows is measured as being  $240^\circ$  then the angle between the two vectors is  $360^\circ - 240^\circ = 120^\circ$

0 1 . 3

Describe what will happen to the angle  $c$  and the magnitude of vector **a** when vector **a** is multiplied by the scalar 2

[2 marks]

0 1 . 4

Describe what will happen to the angle  $c$  and the magnitude of vector **a** when vector **a** is multiplied by the scalar -1

[2 marks]

**Turn over for the next question**

**Turn over ►**

**0 2**

A regular language is a language that can be defined by a regular expression.

**0 2 . 1**

Complete the unshaded cells of **Table 1** to show which of the statements about regular languages are true and which are false.

**Table 1**

| Statement   | True or False? |
|---|----------------|
| All regular languages can be represented using a finite state machine without outputs.                      |                |
| The set of strings defined by a regular language is always finite in size.                                  |                |
| There are some languages which can be represented in Backus-Naur Form (BNF) that are not regular languages. |                |

Copy the contents of the unshaded cells in **Table 1** into the table in your Electronic Answer Document.

**[2 marks]**

**Figure 2** shows a set of Backus-Naur Form (BNF) rules that are used to define a language.

**Figure 2**

```

<sentence> ::= <np><v>
<sentence> ::= <v><np>
<np>      ::= <d><n>
<n>       ::= human | dog | cat | baby
<d>       ::= a | the
<v>       ::= ate | slept | drank | cuddle
<c>       ::= and | but | or

```

**0 2 . 2**

There are two rules in **Figure 2** with `<sentence>` on the left-hand side that are used to define what a valid sentence is in the language.

Write a single rule that could replace these two rules. Your new rule must not change what a valid sentence is.

**[1 mark]**

- 0 2 . 3** Complete the unshaded cells of **Table 2** to show which of the strings are valid sentences for the language defined by the BNF rules in **Figure 2**

**Table 2**

| String         | Valid sentence (Y/N)? |
|----------------|-----------------------|
| cuddle the cat |                       |
| drank a human  |                       |
| the cat slept  |                       |
| cat or dog     |                       |

Copy the contents of the unshaded cells in **Table 2** into the table in your Electronic Answer Document.

**[1 mark]**

- 0 2 . 4** The sentence `dog slept` is not currently valid in the language defined in **Figure 2**

Change the language defined in **Figure 2** by either adding or modifying **exactly one rule** so that `dog slept` is a valid sentence.

You must **not** use the terminals `dog` or `slept` directly in the rule you write/change.

**[1 mark]**

The sentence `the cat slept but the dog drank` is not currently valid in the language defined in **Figure 2**.

To allow this to be a valid sentence either one of the following two additional BNF rules could be used.

```
<sentence> ::= <np><v><c><np><v>
<sentence> ::= <sentence><c><sentence>
```

- 0 2 . 5** State the number of different sentences defined by the rule:

```
<sentence> ::= <np><v><c><np><v>
```

You should only use the rules in **Figure 2** when working out your answer to this question. **Do not** include any additional sentences that your answer to question **02.4** would make valid.

You can get full marks for this question for stating either the number of valid sentences or the full calculation needed to work out the number of valid sentences.

You should show your working.

**[2 marks]****Turn over ►**

**0 2 . 6** State how many more sentences are defined by the rule:

$\langle \text{sentence} \rangle ::= \langle \text{sentence} \rangle \langle c \rangle \langle \text{sentence} \rangle$

than by the rule:

$\langle \text{sentence} \rangle ::= \langle \text{np} \rangle \langle v \rangle \langle c \rangle \langle \text{np} \rangle \langle v \rangle$

**[1 mark]**

**0 3 . 1** There are four athletes taking part in a race. Each athlete is allocated a lane that they run in. There can only be one athlete in each lane and there are four lanes available.

How many different permutations are there for allocating athletes to lanes?

**[1 mark]**

**0 3 . 2** If there are  $n$  athletes taking part in a race and  $n$  lanes available, with each athlete being allocated to one lane and each lane only used by one athlete, how many different permutations are there for allocating athletes to lanes?

**[1 mark]**

**0 3 . 3** An anagram of a string is another string that contains exactly the same number of each character as in the original string but the characters may not all be in the same positions as in the original string. The original string is also an anagram of itself.

If there are  $n$  characters in a string, why will the number of anagrams of the string not always be the same as your answer to question **03.2**?

**[1 mark]**

**0 3 . 4** The travelling salesperson problem (TSP) is when a salesperson has to visit every city in a set of cities and needs to find the shortest route that does this without visiting any city more than once before finally returning to the city that they started from.

The TSP is an example of an intractable problem.

Explain what is meant by an intractable problem.

**[2 marks]**

**Figure 3** lists some time complexities, where  $n$  is the size of the problem input and  $k$  denotes a constant.

**Figure 3**

$O(1)$   
 $O(n^k)$   
 $O(k^n)$   
 $O(n)$   
 $O(\log n)$   
 $O(n \log n)$

**0 3 . 5** Assuming that the time complexities in **Figure 3** are for the most time efficient algorithm that solves a problem, how many of the time complexities in **Figure 3** are for intractable problems?

[1 mark]

**0 3 . 6** State which of the time complexities shown in **Figure 3** is the time complexity of the **linear** search algorithm and explain why it has that time complexity.

[2 marks]

**0 3 . 7** State which of the time complexities shown in **Figure 3** is the time complexity of the **binary** search algorithm and explain why it has that time complexity.

[2 marks]

**0 4** Priority queues and linear queues are examples of data structures.

**0 4 . 1** A data structure can be implemented as a dynamic data structure or as a static data structure.

Discuss the advantages and disadvantages of dynamic data structures compared to static data structures.

[4 marks]

**0 4 . 2** Describe the steps involved in adding an item to a linear queue that has been implemented as a static data structure using an array. Your answer should include a description of how any pointers are used and changed.

[3 marks]

**0 4 . 3** For a priority queue, the process to determine where a new item should be added is more complex than it is with a linear queue.

Describe the steps involved when adding an item to a priority queue, implemented as a static data structure using an array, that were not required when adding an item to a linear queue.

[3 marks]

**Turn over for Section B**

**Turn over ►**

---

**Section B**

You are advised to spend no more than **20 minutes** on this section.

Enter your answers to **Section B** in your Electronic Answer Document.

You **must save** this document at regular intervals.

The question in this section asks you to write program code  
**starting from a new program/project/file.**

You are advised to **save** your program at regular intervals.

---

**0 5**

Write a program that asks the user how many numeric digits they would like to enter and then gets the user to enter that number of numeric digits.

The program should calculate and display the number of times the most frequently entered numeric digit was input.

**Example**

If the user says they are going to enter four digits and then enters the digits 3, 4, 5 and 3, the program should display the value 2 as the most frequently entered digit was 3 and that digit was entered twice.

If more than one numeric digit had the same frequency and was the most frequently entered then instead of displaying the frequency, a message saying "Data was multimodal" should be displayed.

A numeric digit is 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9

You may assume that the number that the user enters to state how many numeric digits there will be and the numeric digits entered by the user are all valid.

**Evidence that you need to provide**

Include the following evidence in your Electronic Answer Document.

**0 5 . 1** Your PROGRAM SOURCE CODE.**[12 marks]****0 5 . 2** SCREEN CAPTURE(S) showing the result of testing the program by entering:

- the number 6 then the numeric digits 0, 1, 2, 1, 2 and 1
- the number 5 then the numeric digits 0, 1, 2, 2 and 1

**[1 mark]**



---

**Turn over for Section C**

**Turn over ►**

---

**Section C**

You are advised to spend no more than **15 minutes** on this section.

Type your answers to **Section C** into your Electronic Answer Document.

You **must save** this document at regular intervals.

These questions refer to the **Preliminary Material** and the **Skeleton Program**, but **do not** require any additional programming.

Refer **either** to the **Preliminary Material** issued with this question paper **or** your electronic copy.

---

**0 6**

This question is about the method `GetIndexOfCompany` in the `Simulation` class.

**0 6 . 1**

What does a return value of `-1` indicate?

[1 mark]

**0 6 . 2**

Some of the program code in this method is unnecessary. Identify which code is not necessary and explain why.

[2 marks]

**0 7**

This question is about the `Outlet` class.

**0 7 . 1**

State the name of a method in the `Outlet` class that uses string concatenation.

[1 mark]

**0 7 . 2**

State the name of a local variable in a method in the `Outlet` class.

[1 mark]

**0 7 . 3**

The `Outlet` class contains some protected attributes. Explain the difference between protected and private attributes.

[1 mark]

**0 7 . 4**

In the `Outlet` class the constant value `100` has been used when calculating the initial value of `DailyCosts` in the constructor method.

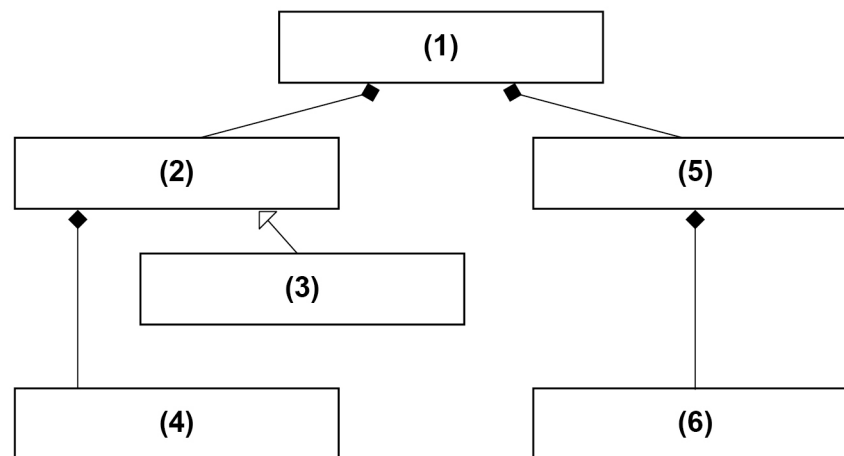
Explain what problem will occur if this value in the constructor is changed and no other changes are made.

[1 mark]

0 8

**Figure 4** shows a partially completed class diagram that describes the relationships between some of the classes used in the Skeleton Program.

**Figure 4**



0 8 . 1

State the identifier of the class that has been labelled (3) in **Figure 4**.

[1 mark]

0 8 . 2

State the identifier of the class that has been labelled (4) in **Figure 4**.

[1 mark]

0 8 . 3

State the identifier of the class that has been labelled (5) in **Figure 4**.

[1 mark]

0 8 . 4

Aggregation, composition and inheritance are three different types of relationship that can exist between classes.

Which of these three types of relationship is **not** shown in **Figure 4**?

[1 mark]

0 9

This question is about the `ProcessDayEnd` method in the `Simulation` class.

Explain how the program decides which company an individual household has decided to use when eating out.

[4 marks]

**Turn over for Section D**

**Turn over ►**

---

## Section D

You are advised to spend no more than **70 minutes** on this section.

Enter your answers to **Section D** in your Electronic Answer Document.

You **must save** this document at regular intervals.

These questions require you to load the **Skeleton Program** and to make programming changes to it.

---

|   |   |
|---|---|
| 1 | 0 |
|---|---|

This question refers to the method `AddCompany` in the `Simulation` class.

The program is to be changed so that it checks that the company name entered is valid. The company must have a name and it cannot be the same as the name of an existing company. The program should keep doing these checks until a valid name for the company has been entered.

### What you need to do

#### Task 1

Modify the method `AddCompany` so that an appropriate error message is displayed if the user presses the Enter key without entering a company name when they are asked to enter the name of the company.

#### Task 2

Modify the method `AddCompany` so that an appropriate error message is displayed if the user enters the name of a company that is already being used for a company in the simulation when they are asked to enter the name of the company.

You may use the method `GetIndexOfCompany` to find out if a company name has already been used, though you do not have to use this method in your answer.

#### Task 3

Test that the changes you have made work:

- run the Skeleton Program
- choose to use a normal size settlement
- choose to use the default companies
- choose the option to add a new company from the menu
- press the Enter key without entering a name for the new company
- type in the name `AQA Burgers` and press the Enter key
- type in the name `In Jest` and press the Enter key.

Note: you will get three marks for your program code if either Task 1 or Task 2 has been successfully completed and you will get five marks if both Task 1 and Task 2 have been successfully completed.

**Evidence that you need to provide**

Include the following evidence in your Electronic Answer Document.

**1 0 . 1** Your PROGRAM SOURCE CODE for the amended method `AddCompany`. **[5 marks]**

**1 0 . 2** SCREEN CAPTURE(S) showing the results of the requested test. **[1 mark]**

**Turn over for the next question**

**Turn over ►**

**1 1**

This question extends the functionality of the simulation by adding a new type of household, an affluent household, that eats out considerably more often than other households.

**What you need to do****Task 1**

Create a new class called `AffluentHousehold` that inherits from the `Household` class. The constructor for this new class should make a call to the constructor of the `Household` class and then set the value of `ChanceEatOutPerDay` to 1

**Task 2**

Modify the `AddHousehold` method in the `Settlement` class so that it creates an `AffluentHousehold` object instead of a `Household` object if the randomly-generated value for `X` is less than 100

**Task 3**

Test that the changes you have made work:

- run the Skeleton Program
- choose to use a normal size settlement
- choose to use the default companies
- then choose the menu option to display the details of the households and check that there is a mixture of both types of household in the settlement.

**Evidence that you need to provide**

Include the following evidence in your Electronic Answer Document.

**1 1 . 1**

Your PROGRAM SOURCE CODE for the amended method `AddHousehold` and the new class `AffluentHousehold`.

**[7 marks]****1 1 . 2**

SCREEN CAPTURE(S) showing the requested test.

**[1 mark]**

---

**Turn over for the next question**

**Turn over ►**

|   |   |
|---|---|
| 1 | 2 |
|---|---|

This question extends the Skeleton Program by adding the facility for a company to obtain a loan to increase their balance.

When a company chooses to get a loan, 10 000 is added to their balance. The company should only be allowed to get a loan if they do not already have a loan that has not been paid back. The user will be asked to specify the daily interest rate for the loan.

The company can pay back all or part of the loan at any time. The program should keep track of the amount of the loan that the company still needs to pay back.

Each time the simulation is advanced by a day, the company will have their balance reduced by an amount equivalent to the interest rate applied to the amount of the loan minus any repayments that have been made. **Figure 5** shows an example of the calculation.

**Figure 5**

A company has repaid 1500 of the loan meaning that it still owes 8500. The interest rate for the loan was 0.0005% and the company's current balance is 4000. The simulation is advanced by a day and the company's balance is reduced by 0.0425 to 3999.9575 (0.0425 is 0.0005% of 8500).

**To get full marks on this question you will need to make use of encapsulation to hide attributes in the `Company` class.**

### What you need to do

#### Task 1

Add attributes to the `Company` class named `LoanBalance` and `InterestRate`. `LoanBalance` should be used to track the amount of the loan that the company needs to pay back. `InterestRate` should be used to store the interest rate at which the loan was taken.

#### Task 2

Modify the `ModifyCompany` method in the `Simulation` class so that there are options on the menu to get a loan and to pay back a loan.

#### Task 3

Modify the Skeleton Program so that if a company has a loan balance of 0 then it can take out a loan. When it does:

- 10 000 should be added to `Balance` and `LoanBalance`
- `InterestRate` should be set to the value specified by the user.

#### Task 4

Modify the Skeleton Program so that when a company chooses to pay back a loan the user is asked how much to pay back and the values of `LoanBalance` and `Balance` are reduced by this amount. You **do not** need to check that the value entered is sensible or that the company has a loan to pay back.



**Task 5**

Modify the `ProcessDayEnd` method in the `Company` class so that the company's daily interest for the loan is subtracted from the company's balance.

**Task 6**

Test that the changes you have made work:

- run the Skeleton Program
- choose to use a normal size settlement
- choose to use the default companies
- give the company AQA Burgers a loan with a percentage interest rate of 0.015
- advance the simulation
- get the company AQA Burgers to pay back 500 of the loan
- advance the simulation
- then choose the menu option to display the details of the companies.

**Evidence that you need to provide**

Include the following evidence in your Electronic Answer Document.

- |  |  |
|--|--|
| <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> | <p>Your PROGRAM SOURCE CODE for the amended parts of your Skeleton Program. You should include the entire code for each method you have changed/created.</p> <p style="text-align: right;"><b>[10 marks]</b></p>   |
| <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">2</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">2</div> | <p>SCREEN CAPTURE(S) showing the requested test. You only need to show the result of choosing the menu option to display the details of the companies. Evidence for the earlier parts of the test is not needed.</p> <p style="text-align: right;"><b>[1 mark]</b></p> |

**Turn over for the next question**

**Turn over ►**

|   |   |
|---|---|
| 1 | 3 |
|---|---|

Currently, the Skeleton Program creates a delivery route for a company based on the order in which a company's outlets have been created – the route starts with the oldest outlet, then goes to the second oldest outlet and so on until all the outlets have been included in the route.

The route produced specifies the order in which the outlets are visited when making deliveries – the longer the route, the higher the delivery cost for the company.

This question extends the Skeleton Program by using a greedy algorithm to try to reduce the total distance of the delivery route. A greedy algorithm is one that takes the choice which seems to be the best each time; it is not guaranteed to find the optimal solution.

**Figure 6** shows how the greedy algorithm should work.

**Figure 6**

The greedy algorithm should:

1. start by adding the oldest outlet (the one in position 0 in `Outlets`) to the route
2. then find the company's outlet that is nearest (has the smallest distance) to the outlet that was last added to the route, and which has not yet been added to the route, and add this to the end of the route
3. repeat step 2 until all the company's outlets have been added to the route.

**Figure 7** shows how the delivery cost is arrived at for the company Paltry Poultry which has four outlets. The outlet numbers (0–3) refer to the index of an outlet within the `Outlets` data structure for that company. This example may help you check your code works correctly.

**Figure 7**

- The oldest outlet (outlet 0) is at coordinates (800,390) within the settlement; this will be the first outlet on the route.
- From this outlet the closest outlet is outlet 2 which is at (820,370) so outlet 2 will be added as the second outlet on the route.
- From outlet 2 the closest outlet not yet in the route is outlet 3 which is at (800,600) so outlet 3 will be added as the third outlet on the route.
- From outlet 3 the closest outlet not yet in the route is outlet 1 which is at (400,390) so outlet 1 will be added as the fourth outlet on the route.
- The correct order that the outlets should be visited is 0, 2, 3, 1
- The delivery cost would be 6.96708

---

**What you need to do****Task 1**

Create a new method `GetOrderedListOfOutlets` in the `Company` class. The new method should return a list that represents the route created using the algorithm in **Figure 6**. Each outlet should be represented in the list by the index of its position in `Outlets`.

**Task 2**

Modify the `CalculateDeliveryCost` method in the `Company` class so that it uses the list returned by the new method `GetOrderedListOfOutlets` instead of the list returned by the method `GetListOfOutlets`.

**Task 3**

Test that the changes you have made work:

- run the **Skeleton Program**
- choose to use a normal size settlement
- choose to use the default companies
- choose the menu option to display the details of the companies.

**Evidence that you need to provide**

Include the following evidence in your Electronic Answer Document.

- |  |  |                   |
|--|--|-------------------|
| <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> | Your PROGRAM SOURCE CODE for the amended method <code>CalculateDeliveryCost</code> and for the new method <code>GetOrderedListOfOutlets</code> . | <b>[11 marks]</b> |
| <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">1</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">3</div> <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">2</div> | SCREEN CAPTURE(S) showing the test described in <b>Task 3</b> . The screen capture should show the delivery cost for <b>AQA Burgers</b> .        | <b>[1 mark]</b>   |

**END OF QUESTIONS**

---

**There are no questions printed on this page**

**Copyright information**

For confidentiality purposes, all acknowledgements of third-party copyright material are published in a separate booklet. This booklet is published after each live examination series and is available for free download from [www.aqa.org.uk](http://www.aqa.org.uk).

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and AQA will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team.

Copyright © 2020 AQA and its licensors. All rights reserved.



2 0 6 A 7 5 1 7 / 1