GCSE

C500U20-1

S19-C500U20-1

Part of WJEC

# COMPUTER SCIENCE – Component 2
## Computational Thinking and Programming

THURSDAY, 16 MAY 2019 – AFTERNOON

2 hours

### ADDITIONAL MATERIALS

Your computer should be pre-installed with text editing software, a word processing package and a functional copy of version 2.4.2 of the Greenfoot IDE.

### INSTRUCTIONS TO CANDIDATES

You will need to enter your answers to questions 1, 3, 4, 5, 6 and 7 within the electronic answer document provided.

You will need to create a new plain text file to answer question 2.

You will complete the work for question 8 within the Greenfoot IDE.

Carry out all tasks and make sure that you check your work carefully to ensure that the work you produce is accurate and correct.

Save your work regularly.

### INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

You are reminded of the need for good English and orderly, clear presentation in your answers.

The total number of marks available for this examination is 60.

Questions 3, 5, 6, 7 and 8 will require you to draw on your knowledge from multiple areas of your course of study.

**1.** State the HTML tags needed for each of the following: [3]

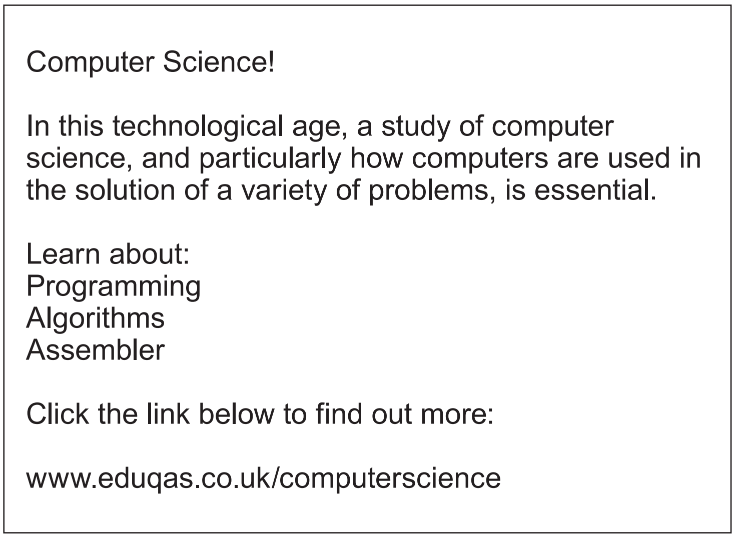
*(a)* Centre align a section of text.

*(b)* Setup the most important heading.

*(c)* Indicate a section of text that is quoted from another source.

Enter your answers in the electronic answer document.

**2.** A draft design for an HTML web page is shown below. [10]



Computer Science!

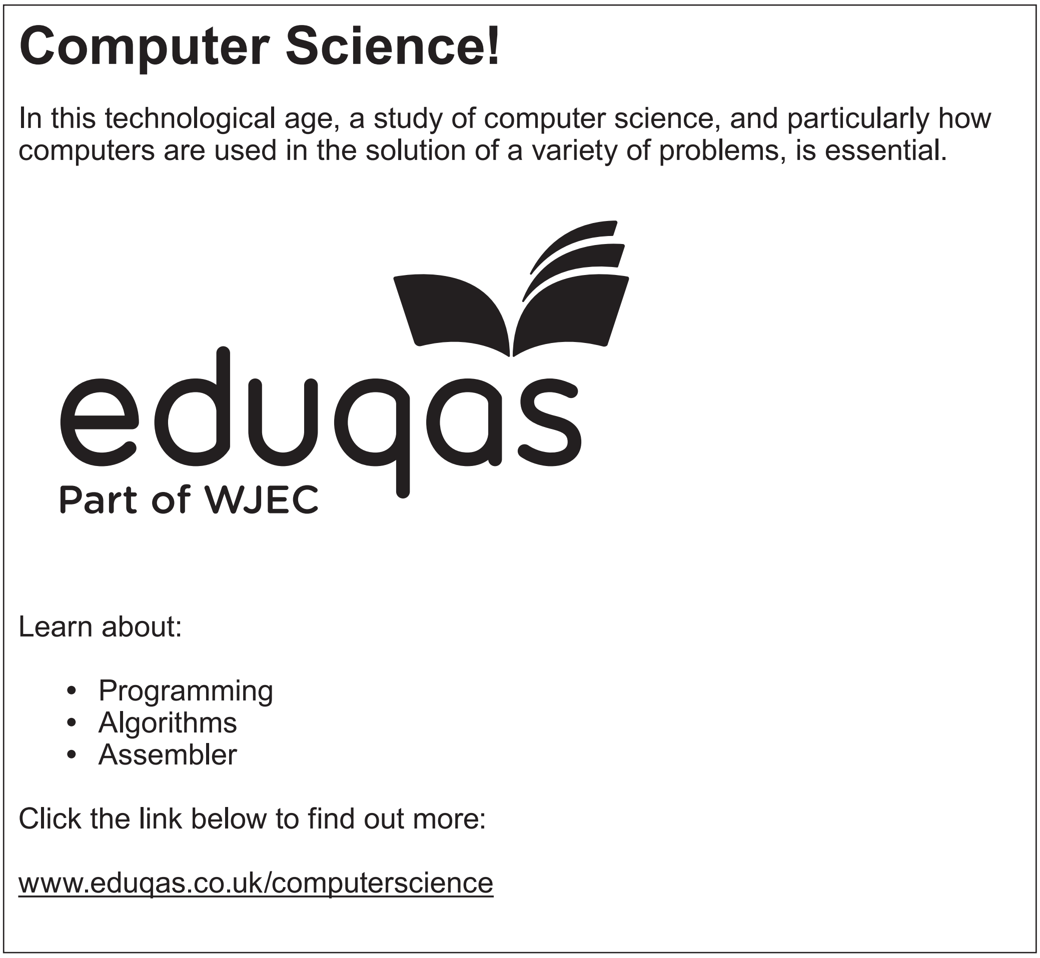
In this technological age, a study of computer science, and particularly how computers are used in the solution of a variety of problems, is essential.

Learn about:
Programming
Algorithms
Assembler

Click the link below to find out more:

www.eduqas.co.uk/computerscience

The design was then improved using various HTML tags to provide the formatting and content shown below.



# Computer Science!

In this technological age, a study of computer science, and particularly how computers are used in the solution of a variety of problems, is essential.

Learn about:

- Programming
- Algorithms
- Assembler

Click the link below to find out more:

www.eduqas.co.uk/computerscience

Copy the text from the electronic answer document into a basic text editor.

Insert the HTML tags that would be needed to display the content and formatting shown in the improved design.

The image file you require is called `eduqas.jpg`

The page title should be set to `CompSci`

Save your new web page as `finalCompSci.txt`

**3.** An assembly language program is required which can subtract one number from another.

Write a program in assembly language to allow the user to input two numbers, store the two numbers in two declared variables, subtract one from the other, and output the resulting number. [9]

Enter your answer in the electronic answer document.

**4.** *(a)* Describe what is meant by the term *algorithm*. [2]

*(b)* Describe what is meant by the term *iteration*. [2]

*(c)* Describe what is meant by the term *rogue value*. [1]

Enter your answers in the electronic answer document.

**5.** Below is an algorithm:

```
dataList[8] is integer array
dataList = [1, 3, 21, 45, 57, 17, 34, 65]
              {Populate the array}

Declare linearSearch(dataList, searchItem)
 position is integer
 found is Boolean
 set position = 1
 set found = FALSE

 do while position < length(dataList) AND found = FALSE
  if dataList[position] = searchItem then
   found = TRUE
  else
   position = position + 1
  end if
 loop

End Subroutine
```

*(a)* From the algorithm above identify an example of: [5]

(i) selection

(ii) incrementation

(iii) assignment

(iv) declaration

(v) annotation

Enter your answers in the electronic answer document.

*(b)* Complete the table given in the electronic answer document to show the values of the variable found for each of the given inputs. [4]

| Input | found = |
|---|---|
| linearSearch(dataList, 2) | |
| linearSearch(dataList, 3) | |
| linearSearch(dataList, 54) | |
| linearSearch(dataList, 65) | |

Complete the table in the electronic answer document.

**6.** An algorithm intended to calculate the area of a circle is shown below.

The algorithm accepts a single input; the radius, and outputs the area.
The area of a circle is calculated by multiplying Pi by the radius squared.

```
1   set Pi = 3.142
2   radius is real
3   . . .
4
5   Declare FindArea
6     {procedure to calculate the area of a circle}
7
8    . . .
9
10  End Subroutine
11
12  startMainProg
13   output “Type in the radius”
14   . . .
15
16   call FindArea
17
18   output “The area is:”
19   . . .
20  endMainProg
```

Lines 3, 8, 14 and 19 are missing from the algorithm above.

Using four of the lines of code below, complete this algorithm in the electronic answer document. [4]

- `else is TRUE`
- `output area`
- `input radius`
- `input flag`
- `area  = Pi * radius * radius`
- `End Subroutine`
- `area is real`

**7.** Open the Greenfoot world `WJECSnake7` and familiarise yourself with its contents.

From the Greenfoot world `WJECSnake7` identify an example of each of the following: [5]

*(a)* superclass

*(b)* class

*(c)* private property

*(d)* parameter

*(e)* comment.

Enter your answer in the electronic answer document.

**8.** Open the Greenfoot world `WJECSnake8` and familiarise yourself with its contents. Complete the world as instructed below: [15]

*(a)* Populate the world with a **snake**, **meerkat** and at least two **eggs**.

*(b)* Edit the **meerkat** and **egg** classes so that they turn and move at random.

*(c)* Edit the **snake** code so that it moves at an appropriate speed in the direction of the arrow keys when pressed.

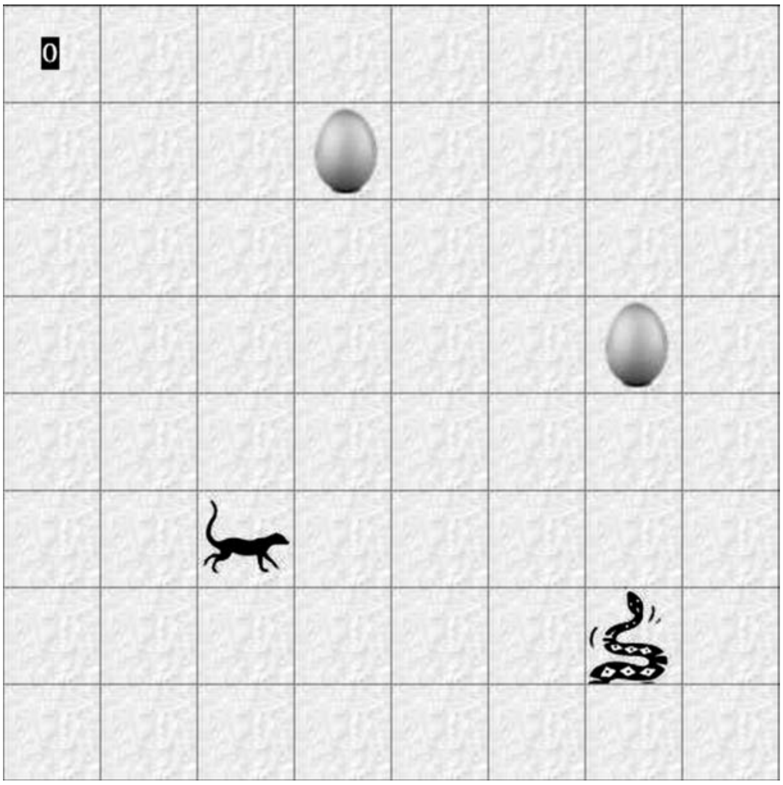
*(d)* Edit the **snake** code so that it “eats” an **egg** when they collide (removes the **egg** from the world).

*(e)* Add a sound which will play every time the **snake** “eats” an **egg**.

*(f)* Add a **counter**. Edit the code so that the **counter** displays how many **eggs** have been “eaten” by the snake.

*(g)* Edit the code so that the **counter** loses a point (1 point is deducted) if the **meerkat** collides with the **egg**.

*(h)* Save your completed world as `FinalSnake8`



**END OF PAPER**