GCSE – NEW

3500U20-1

S19-3500U20-1

COMPUTER SCIENCE
UNIT 2: Computational Thinking and Programming

THURSDAY, 16 MAY 2019 – AFTERNOON

2 hours

3500U201
01

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 8 into 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 7 and 9 within the Greenfoot IDE.

Carry out all tasks and save your work regularly.

INFORMATION FOR CANDIDATES

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

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

1. State the HTML tags needed to: [4]

(a) indicate where the header metadata should be stored.

(b) add a list item.

(c) define an image.

(d) specify a section of text that is quoted from another source.

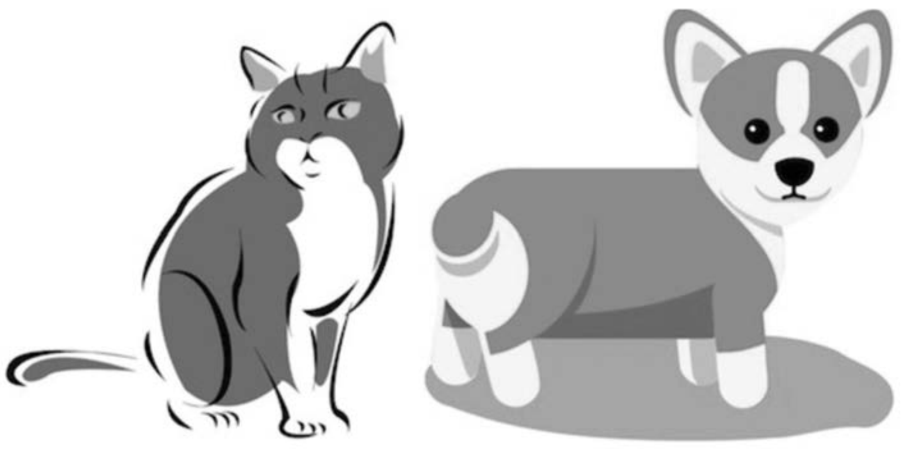
Enter your answers into the electronic answer document.

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

> TrackMyPetCare.com
>
> The new online system for keeping track of all your pets' needs, including:
>
> - Vet check-ups
> - Booster dates
> - Birthdays and other anniversaries!
>
> Click the link below to find out more:
>
> www.TrackMyPetCare.com

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

TrackMyPetCare.com



The new online system for keeping track of all your pets' needs, including:

- Vet check-ups
- Booster dates
- Birthdays and other anniversaries!

Click the link below to find out more:

www.TrackMyPetCare.com

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:	`pets.jpg`
The page title should be set to:	`Track My Pet Care`
Save your new web page as:	`finalPet.txt`

3. State the assembly language mnemonic to: [4]

(a) input a value.

(b) create a data definition.

(c) branch the program execution.

(d) subtract a value from a register.

Enter your answers into the electronic answer document.

4. Below is an algorithm:

```
total is integer
set total = 0
Declare Subroutine CountUp
  counter is integer
  set counter = 0
  output “About to count”
  do
    counter = counter + 1
    total = total + counter
    output “Count is ”, counter
  while counter < 3      {Note: the loop has ended here}
  output “Count complete, total is ”, total
End Subroutine
```

From the algorithm identify an example of: [4]

(a) a local variable

(b) a global variable

(c) annotation

(d) assignment

Enter your answers into the electronic answer document.

5. Below is an algorithm:

```
outValue is integer
set outValue = 0

Declare Subroutine Multi

 for i = 1 to 3
  for j = 1 to 3
   outValue = i * j
   output outValue
  next j
 next i

End Subroutine
```

Complete the table in the electronic answer document to show all the outputs of this algorithm. [9]

6. An algorithm is required to help scientists monitor the level of a pollutant in a river. They take four readings of the level of pollutant in the river then use a computer to analyse the data. The value of each reading will be an integer in a range from 1 - 10.

The algorithm should:

- accept the input of each reading
- output the total of all the numbers entered
- output the mean of all the numbers entered
- output the largest number entered
- output the smallest number entered

An example of the ***input*** and output required is shown below.

```
Enter reading: 6

Enter reading: 3

Enter reading: 2

Enter reading: 4


Total: 15

Mean: 3.75

Largest: 6

Smallest: 2
```

Write an algorithm to meet these requirements. Enter your algorithm into the electronic answer document. [6]

7. A pet shop would like a new scenario created using the Java programming language within the Greenfoot environment. [5]

(a) Create a new world in the Greenfoot environment called **tank**. Set the background image within this world to a 9 x 9 grid using the image `water.jpg`

(b) Create a new class called **fish** and set the image of this class to `fish.jpg`
Populate the world with two **fish**.

(c) Create a new class called **shark** and add code to this class to allow the **shark** to move and turn at random. Set the image of this class to `shark.jpg`
Populate the world with two **sharks**.

(d) Create a new class called **crab**. Add code to this class to allow the **crab** to move only left and right at random. Set the image of this class to `crab.jpg`
Populate the world with a **crab**.

(e) Save your completed world as `finalAquarium7`

All of the images you require are in the `Aquarium\images` folder.

8. Open the Greenfoot world `WJECAnts8` and familiarise yourself with its contents. [5]

From the Greenfoot world `WJECAnts8` identify an example of a:

(a) superclass

(b) class

(c) private property

(d) comment

(e) method which is automatically run in each frame

Enter your answers into the electronic answer document.

9. Open the Greenfoot world `WJECAnts9` and familiarise yourself with its contents. Complete the world as instructed below: [13]

(a) Populate the world with an **ant**, a **waterDrop** and at least two instances of **sugarCube**.

(b) Edit the **waterDrop** and **sugarCube** objects so that they turn and move around the world at random.

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

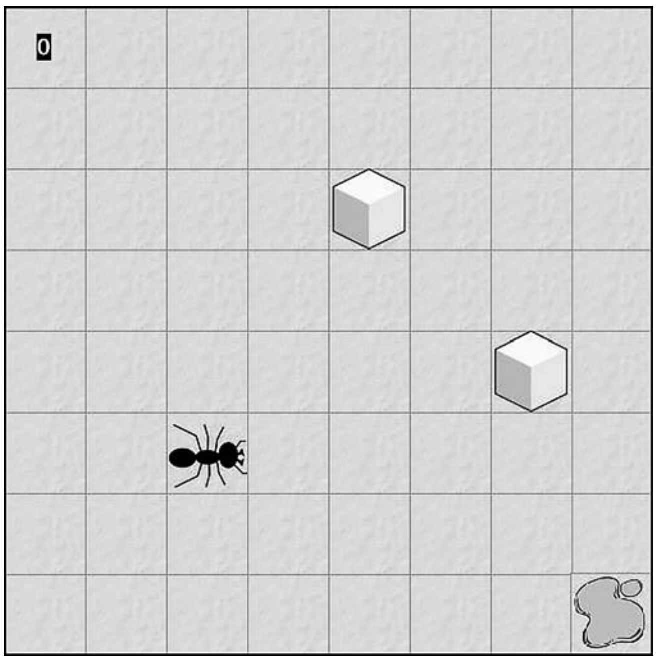
(d) Edit the **ant** object so that it "eats" a **sugarCube** when they collide (removes the **sugarCube** from the world).

(e) Add a sound which will play every time the **ant** "eats" a **sugarCube**.

(f) Add a **counter**. Edit the code so that the **counter** displays how many **sugarCubes** have been "eaten".

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

(h) Save your completed world as `finalAnts9`



END OF PAPER

BLANK PAGE

BLANK PAGE