GCSE

3500U20-1

Z22-3500U20-1

THURSDAY, 16 JUNE 2022 – MORNING

COMPUTER SCIENCE

UNIT 2: Computational Thinking and Programming

2 hours

ADDITIONAL MATERIALS

You will require the electronic answer document for this examination and files for questions 5, 6 and 7 all of which should be pre-installed on your examination account.

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

INSTRUCTIONS TO CANDIDATES

You will need to enter your answers to questions 2, 3, 4, and 6 within the electronic answer document provided.

You will need to create a new plain text file to answer question 1.

You will complete the work for questions 5 and 7 within the Greenfoot IDE.

Carry out all tasks and save your work regularly.

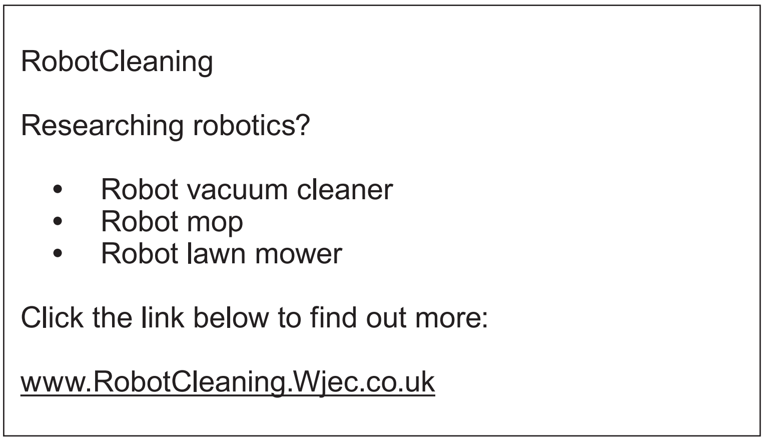
INFORMATION FOR CANDIDATES

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.

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

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



RobotCleaning

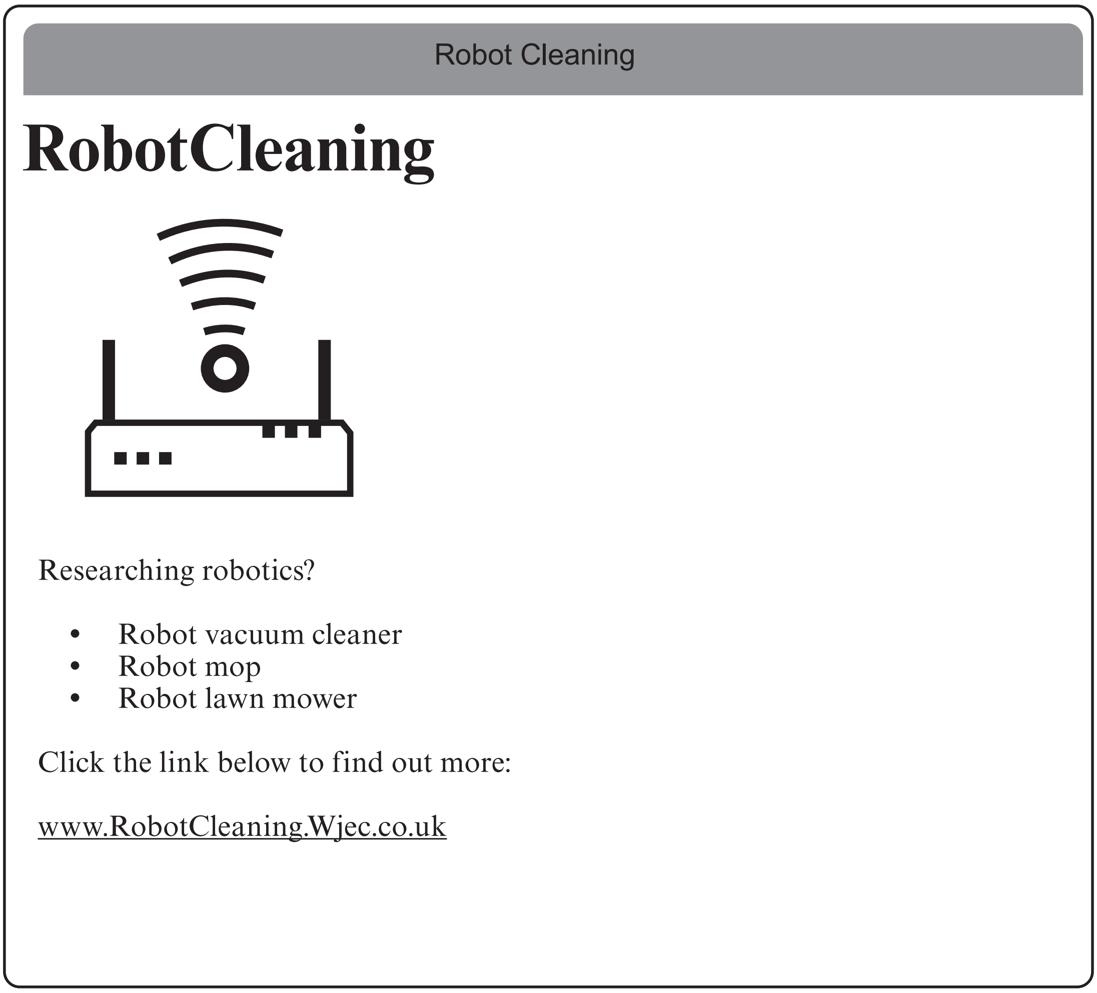
Researching robotics?

- Robot vacuum cleaner
- Robot mop
- Robot lawn mower

Click the link below to find out more:

www.RobotCleaning.Wjec.co.uk

The format, content and layout of the design has been improved, as shown.



Robot Cleaning

RobotCleaning

Researching robotics?

- Robot vacuum cleaner
- Robot mop
- Robot lawn mower

Click the link below to find out more:

www.RobotCleaning.Wjec.co.uk

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

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

The image file you require is called:	`robot.jpg`
The page title should be set to:	`RobotCleaning`
Save your new web page as:	`RobotCleaning.txt`

2. (a) State the assembly language mnemonic for each of the following: [4]

(i) to output a value.

(ii) to load a value.

(iii) to store a value.

(iv) to add two values together.

Enter your answers in the electronic answer document.

(b) Write an assembly language program to load two values and add them together. [4]

Enter your answers in the electronic answer document.

3. Complete the truth tables in the electronic answer document. [4]

A	B	A AND B

A	NOT A

4.

```
total is integer
set total = 0
Declare Subroutine Add
  counter is integer
  set counter = 0
  output “About to add”
  output counter
  do
    counter = counter + 1
    total = total + counter
    output counter
  while counter < 2
  output “The total is ” total
End Subroutine
```

(a) Give all the outputs of the algorithm. [5]

(b) Identify an example of the following from the algorithm: [4]

(i) a global variable.

(ii) a local variable.

(iii) assignment.

(iv) iteration.

(c) An algorithm is required to help analyse a 3 hour survey of the number of cars using a road.

The algorithm should:

- accept the input of the number of cars per hour
- output the total number of cars over 3 hours
- output the mean number of cars each hour
- output the largest number of cars in an hour
- output the smallest number of cars in an hour

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

```
Enter reading: 65

Enter reading: 37

Enter reading: 24


Total: 126

Mean: 42

Largest: 65

Smallest: 24
```

Write an algorithm to meet these requirements. Enter your algorithm in the electronic answer document. [7]

5. A garden centre would like a new scenario created in the Java programming language within the Greenfoot environment. The garden centre will use the scenario as a screen saver. [5]

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

(b) Create a new class called **Leaves** and set the image of this class to `leaf.jpg`

(c) Populate the world with six **Leaves**.

(d) Enter code into the **Leaves** class so that the leaves turn and move at random (as if blowing in the wind).

(e) Save your completed world as `finalLeaves`

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

6. Open the Greenfoot world `WJECTrees6` and familiarise yourself with its contents. [5]

From the Greenfoot world `WJECTrees6` identify an example of a:

(a) superclass

(b) class which inherits from World

(c) private property

(d) method

(e) parameter

Enter your answers in the electronic answer document.

7. Open the Greenfoot world `WJECTrees7` and familiarise yourself with its contents. Complete the world as instructed below:

(a) Populate the world with a **tree**, **sun** and at least two instances of a **waterDrop**. [3]

(b) Edit the **waterDrop** and **sun** objects so that they turn and move around the world at random. [1]

(c) Edit the **tree** object so that it moves at an appropriate speed in the direction of the arrow keys when pressed. [2]

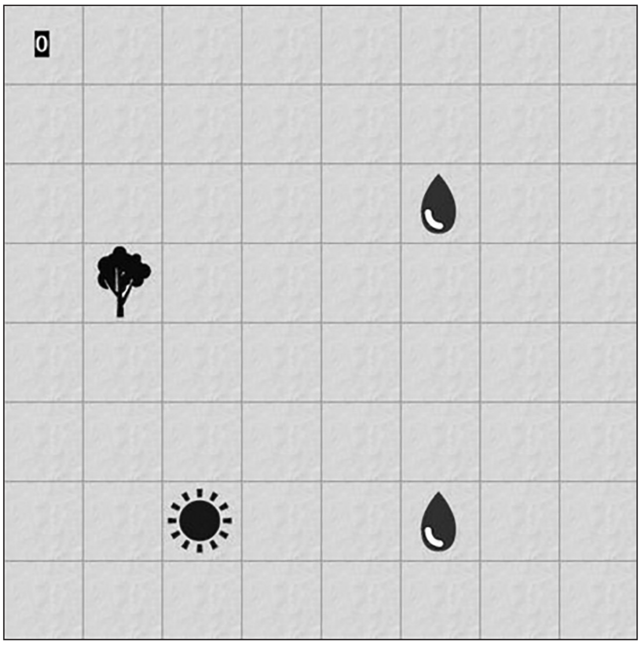
(d) Edit the **tree** object so that it “drinks” a **waterDrop** when they collide (removes the **waterDrop** from the world). [1]

(e) Add a sound which will play every time the **tree** “drinks” a **waterDrop**. [1]

(f) Add a **counter**. Edit the code so that the **counter** displays how many **waterDrops** have been “drunk” by the **tree**. [2]

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

(h) Save your completed world as `FinalWJECTrees7`. [1]



END OF PAPER

BLANK PAGE