



**GCSE**

C500U20-1



**FRIDAY, 6 NOVEMBER 2020 – AFTERNOON**

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

2 hours

**ADDITIONAL MATERIALS**

You will require the electronic answer document for this examination and files for questions 1, 6 and 7.

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 5 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 question 6 and 7 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 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.

Certain questions will require you to draw on your knowledge from multiple areas of your course of study.

1. A draft design for an HTML web page is shown.

[10]

## Mobile Device Management

In this Mobile First age it is vital that all company devices are managed.

MDM software allows devices to be:

Locked

Wiped

Tracked

From anywhere in the world!

Click the link below to find out more:

[www.manageourmobiles.com](http://www.manageourmobiles.com)

“Mobile device management is a must for any company or organisation” – CEO

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

Manage mobile devices ✕

## Mobile Device Management

In this Mobile First age it is vital that all company devices are managed.



MDM software allows devices to be:

- Locked
- Wiped
- Tracked

From **anywhere** in the world!

Click the link below to find out more:

[www.manageourmobiles.com](http://www.manageourmobiles.com)

“Mobile device management is a must for any company or organisation”

- CEO

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 quotation from the CEO should be set as a blocked quote.**

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

The page title should be set to `Manage mobile devices`

Save your new web page as `finalmobiles.txt`

2. (a) State the assembly language mnemonic for each of the following: [3]

- (i) to output a value.
- (ii) to store a value.
- (iii) to subtract a value.

(b) The table below gives examples of different types of data to be stored. [6]

Example:	Most appropriate data type:
"Hello"	
1.1	
TRUE	
465	
L	
"The quick brown fox"	

Complete the table in the electronic answer document to show the most appropriate data type for each example.

3. (a) Describe what is meant by the following: [5]

- (i) scope of a variable.
- (ii) lifetime of a variable.

Enter your answers in the electronic answer document.

(b) Below is an algorithm:

```
1 multi is integer
2 input multi
3 Declare TimesUp
4 counter is integer
5 set counter = 1
6 repeat
7   output "Counter is " counter
8   multi = multi * counter
9   output "Multi is " multi
10  counter = counter + 1
11 until counter = 3
12 End Subroutine
```

Complete the table in the electronic answer document to show the outputs of this algorithm where **the input multi is set to 2**. [4]

4. Below is an algorithm intended to allow a user to log in to a computer program.

```

1  Declare LoginScreen
2  username is string
3  password is string
4  counter is integer
5  loggedIn is boolean
6  set loggedIn = FALSE
7  set counter = 0
8  do
9   output "Type in username"
10  input username
11  output "Type in password"
12  . . .
13  if username = "User1" AND password = "Pass1" then
14   output "Username and password correct"
15   . . .
16  else
17   output "Please try again"
18   . . .
19  end if
20 loop until counter = 3
21 . . .

```

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

```

End Subroutine

input password

set loggedIn = TRUE

set bypassed = FALSE

loop until counter = 3

counter = counter + 1

```

(b) State the purpose of each line you have used to complete this algorithm. [4]

Complete the table in the electronic answer document.

5. Demonstrate how the following data would be sorted in an array using a bubble sort algorithm.

9	5	10	2
---	---	----	---

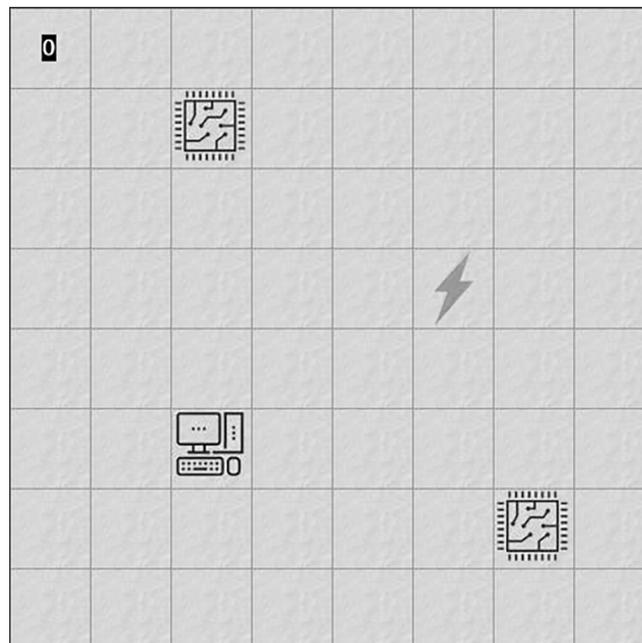
Show how the order of the numbers would change at each stage of the sort.  
Enter your answer in the table in the electronic answer document.

[4]

6. A computer hardware store would like a new scenario created in the Java programming language within the Greenfoot environment. The store will use the scenario as a screen saver. [5]
- (a) Create a new world in the Greenfoot environment called **Advert**. Set the background image within this world to a  $9 \times 9$  grid using the image `cell.jpg`
  - (b) Create a new class called **Phones** and set the image of this class to `phones.jpg`
  - (c) Populate the world with six **Phones**.
  - (d) Enter code into the **Phones** class to make the phones turn and move slowly at random.
  - (e) Save your completed world as *FinalAdvert*.

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

7. Open the Greenfoot world `WJECStatic7` and familiarise yourself with its contents. Complete the world as instructed below: [15]
- Populate the world with a **Computer**, **Static** and at least two instances of a **Chip**.
  - Edit the **Static** and **Chip** classes so that they turn and move around the world at random.
  - Edit the **Computer** code so that it moves at an appropriate speed in the direction of the arrow keys when pressed.
  - Edit the **Computer** code so that it “collects” a **Chip** when they collide (removes the **Chip** from the world).
  - Add a sound which will play every time the **Computer** “collects” a **Chip**.
  - Add a **Counter**. Edit the code so that the **Counter** displays how many **Chips** have been “collected” by the **Computer**.
  - Edit the code so that the **Counter** loses a point (1 point is deducted) if the **Static** collides with the **Chip**.
  - Edit the **Static** or **Chip** code so that the **Chip** is “fried” when the **Static** and **Chip** collide (removes the **Chip** from the world).
  - Save your completed world as `FinalStatic7`



**END OF PAPER**