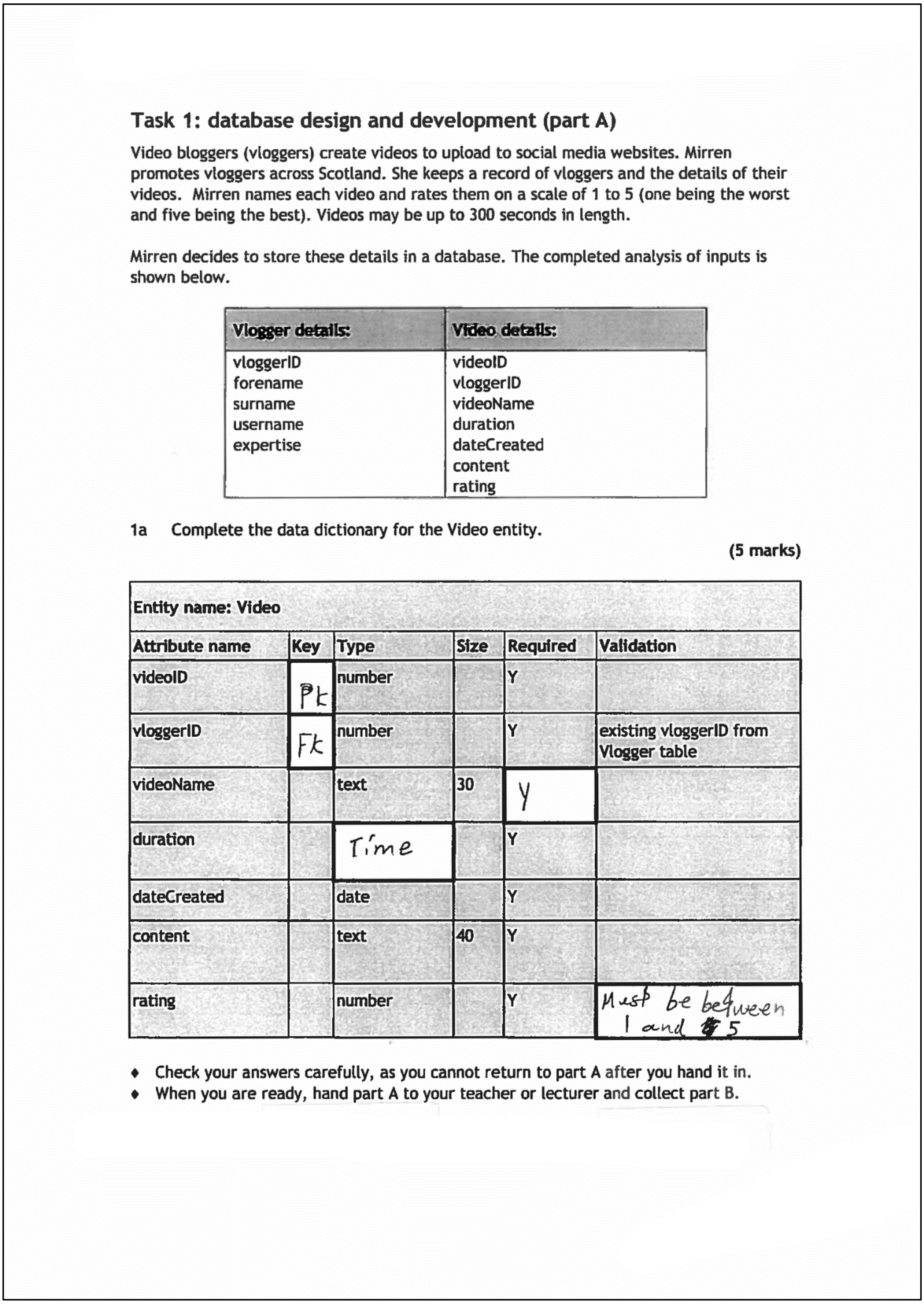
# Candidate 2 evidence

## Database design and development



### Task 1: database design and development (part A)

Video bloggers (vloggers) create videos to upload to social media websites. Mirren promotes vloggers across Scotland. She keeps a record of vloggers and the details of their videos. Mirren names each video and rates them on a scale of 1 to 5 (one being the worst and five being the best). Videos may be up to 300 seconds in length.

Mirren decides to store these details in a database. The completed analysis of inputs is shown below.

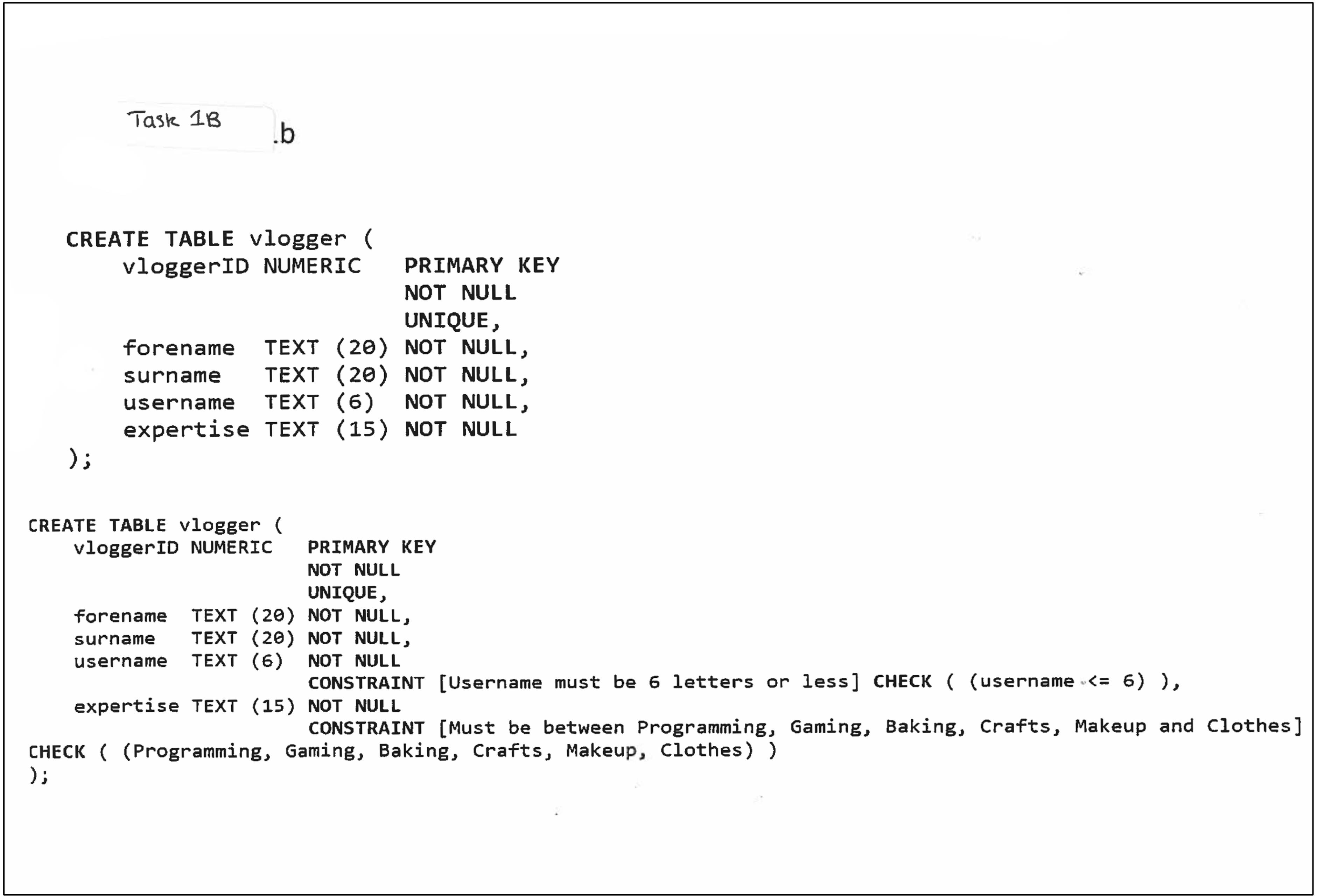
| Vlogger details: | Video details: |
|---|---|
| vloggerID<br>forename<br>surname<br>username<br>expertise | videoID<br>vloggerID<br>videoName<br>duration<br>dateCreated<br>content<br>rating |

1a Complete the data dictionary for the Video entity.

**(5 marks)**

| Entity name: Video | | | | | |
|---|---|---|---|---|---|
| **Attribute name** | **Key** | **Type** | **Size** | **Required** | **Validation** |
| videoID | Pk | number | | Y | |
| vloggerID | Fk | number | | Y | existing vloggerID from Vlogger table |
| videoName | | text | 30 | y | |
| duration | | Time | | Y | |
| dateCreated | | date | | Y | |
| content | | text | 40 | Y | |
| rating | | number | | Y | Must be between 1 and 5 |

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.



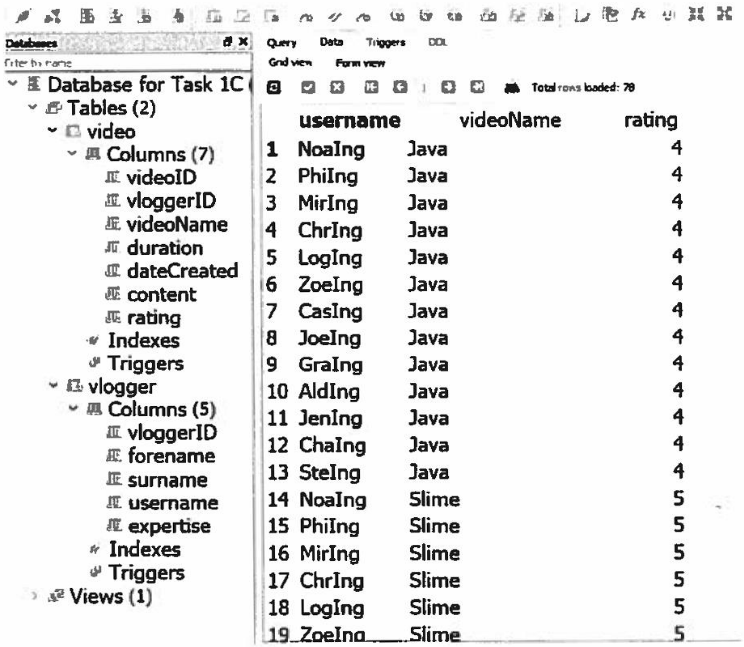
Task 1B .b

```
CREATE TABLE vlogger (
    vloggerID NUMERIC   PRIMARY KEY
                        NOT NULL
                        UNIQUE,
    forename  TEXT (20) NOT NULL,
    surname   TEXT (20) NOT NULL,
    username  TEXT (6)  NOT NULL,
    expertise TEXT (15) NOT NULL
);
```

```
CREATE TABLE vlogger (
    vloggerID NUMERIC   PRIMARY KEY
                        NOT NULL
                        UNIQUE,
    forename  TEXT (20) NOT NULL,
    surname   TEXT (20) NOT NULL,
    username  TEXT (6)  NOT NULL
                        CONSTRAINT [Username must be 6 letters or less] CHECK ( (username <= 6) ),
    expertise TEXT (15) NOT NULL
                        CONSTRAINT [Must be between Programming, Gaming, Baking, Crafts, Makeup and Clothes]
CHECK ( (Programming, Gaming, Baking, Crafts, Makeup, Clothes) )
);
```

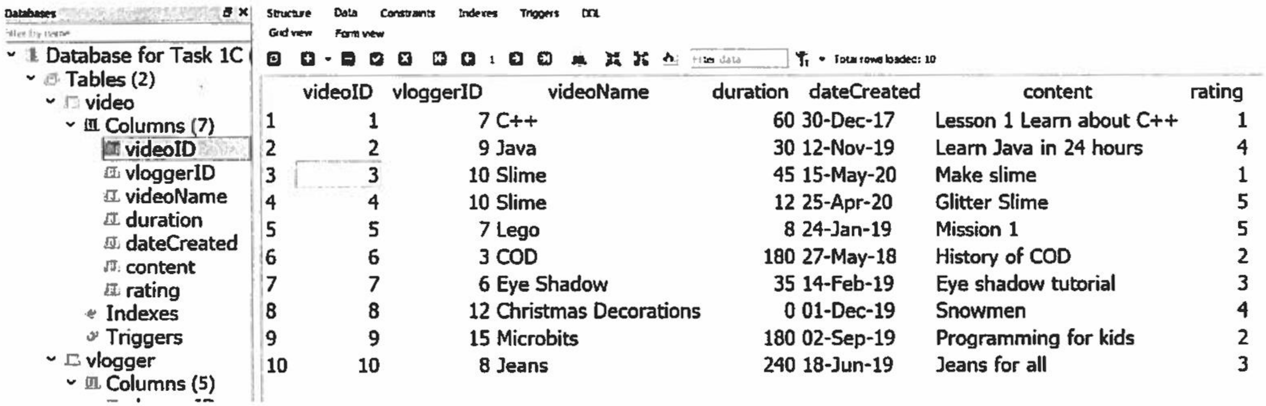
i)

```
CREATE VIEW [The Best Videos] AS
    SELECT vlogger.username,
           video.videoName,
           video.rating
      FROM vlogger,
           video
     WHERE (video.rating >= 3);
```

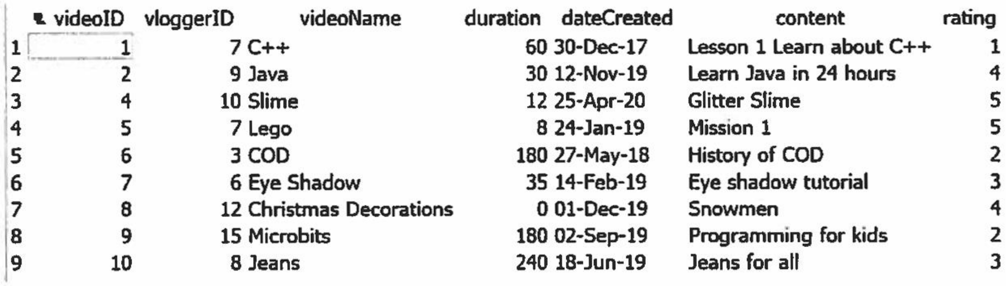


| | username | videoName | rating |
|---|---|---|---|
| 1 | NoaIng | Java | 4 |
| 2 | PhiIng | Java | 4 |
| 3 | MirIng | Java | 4 |
| 4 | ChrIng | Java | 4 |
| 5 | LogIng | Java | 4 |
| 6 | ZoeIng | Java | 4 |
| 7 | CasIng | Java | 4 |
| 8 | JoeIng | Java | 4 |
| 9 | GraIng | Java | 4 |
| 10 | AldIng | Java | 4 |
| 11 | JenIng | Java | 4 |
| 12 | ChaIng | Java | 4 |
| 13 | SteIng | Java | 4 |
| 14 | NoaIng | Slime | 5 |
| 15 | PhiIng | Slime | 5 |
| 16 | MirIng | Slime | 5 |
| 17 | ChrIng | Slime | 5 |
| 18 | LogIng | Slime | 5 |
| 19 | ZoeIng | Slime | 5 |

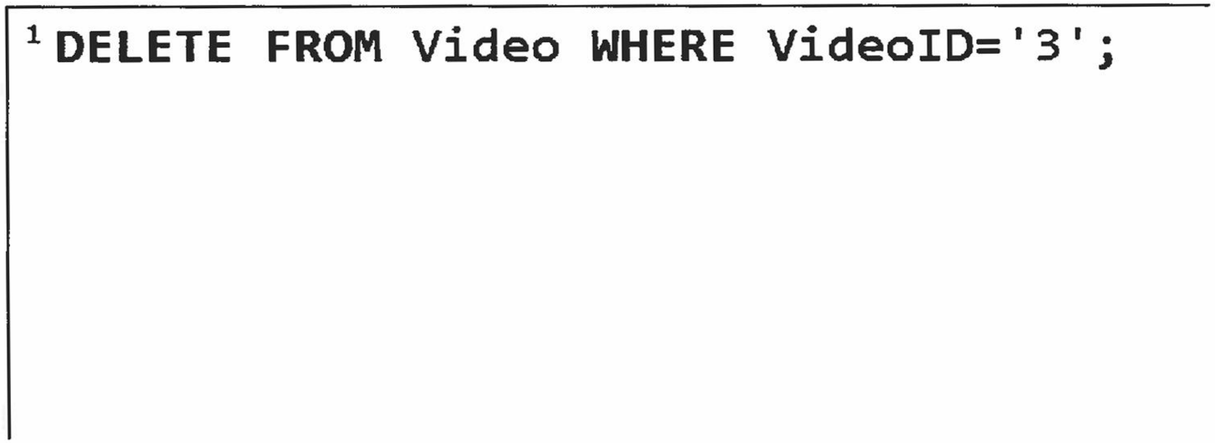
1c ii)



| | videoID | vloggerID | videoName | duration | dateCreated | content | rating |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 7 | C++ | 60 | 30-Dec-17 | Lesson 1 Learn about C++ | 1 |
| 2 | 2 | 9 | Java | 30 | 12-Nov-19 | Learn Java in 24 hours | 4 |
| 3 | 3 | 10 | Slime | 45 | 15-May-20 | Make slime | 1 |
| 4 | 4 | 10 | Slime | 12 | 25-Apr-20 | Glitter Slime | 5 |
| 5 | 5 | 7 | Lego | 8 | 24-Jan-19 | Mission 1 | 5 |
| 6 | 6 | 3 | COD | 180 | 27-May-18 | History of COD | 2 |
| 7 | 7 | 6 | Eye Shadow | 35 | 14-Feb-19 | Eye shadow tutorial | 3 |
| 8 | 8 | 12 | Christmas Decorations | 0 | 01-Dec-19 | Snowmen | 4 |
| 9 | 9 | 15 | Microbits | 180 | 02-Sep-19 | Programming for kids | 2 |
| 10 | 10 | 8 | Jeans | 240 | 18-Jun-19 | Jeans for all | 3 |

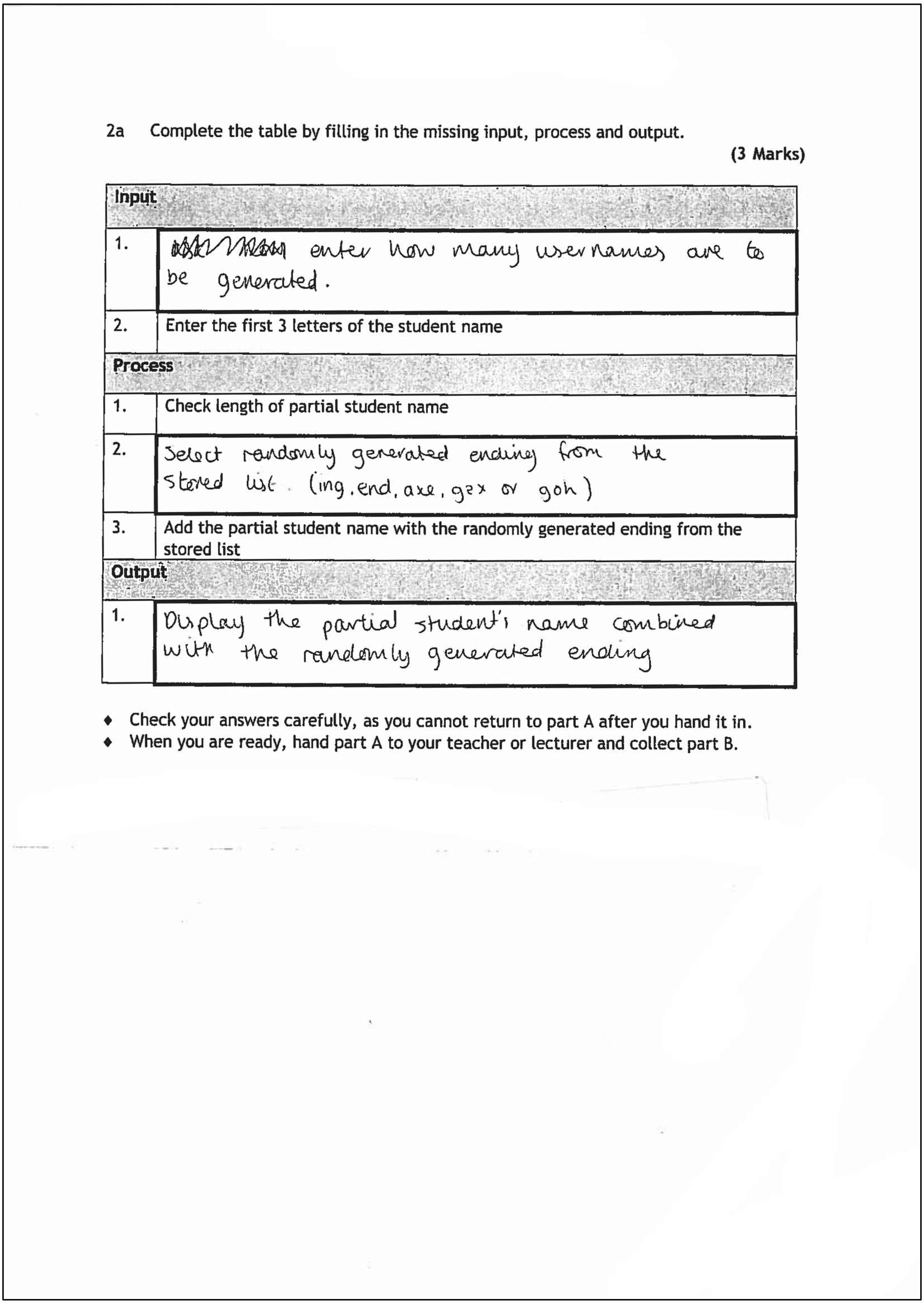


| | videoID | vloggerID | videoName | duration | dateCreated | content | rating |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 7 | C++ | 60 | 30-Dec-17 | Lesson 1 Learn about C++ | 1 |
| 2 | 2 | 9 | Java | 30 | 12-Nov-19 | Learn Java in 24 hours | 4 |
| 3 | 4 | 10 | Slime | 12 | 25-Apr-20 | Glitter Slime | 5 |
| 4 | 5 | 7 | Lego | 8 | 24-Jan-19 | Mission 1 | 5 |
| 5 | 6 | 3 | COD | 180 | 27-May-18 | History of COD | 2 |
| 6 | 7 | 6 | Eye Shadow | 35 | 14-Feb-19 | Eye shadow tutorial | 3 |
| 7 | 8 | 12 | Christmas Decorations | 0 | 01-Dec-19 | Snowmen | 4 |
| 8 | 9 | 15 | Microbits | 180 | 02-Sep-19 | Programming for kids | 2 |
| 9 | 10 | 8 | Jeans | 240 | 18-Jun-19 | Jeans for all | 3 |



```
DELETE FROM Video WHERE VideoID='3';
```

## Software design and development



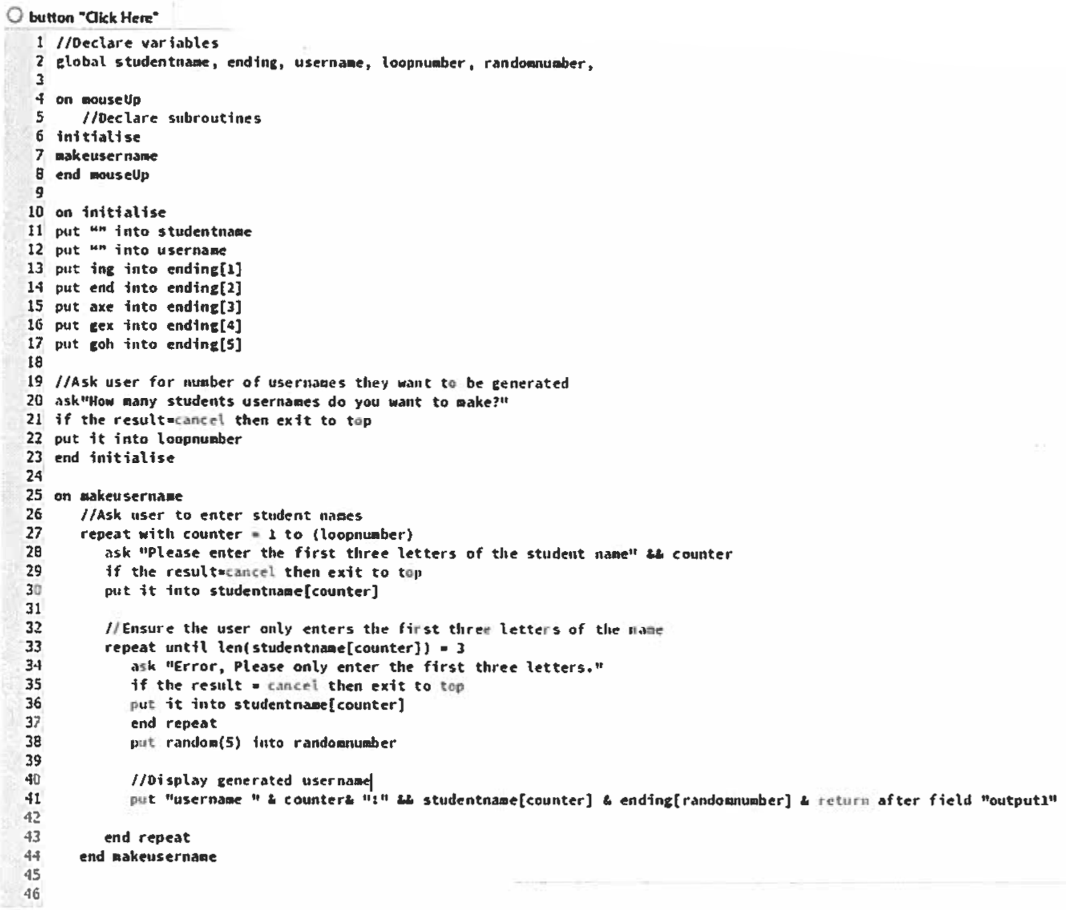
2a Complete the table by filling in the missing input, process and output.

**(3 Marks)**

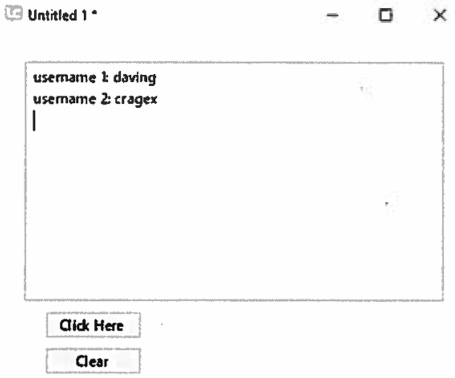
| | |
|---|---|
| **Input** | |
| 1. | enter how many usernames are to be generated. |
| 2. | Enter the first 3 letters of the student name |
| **Process** | |
| 1. | Check length of partial student name |
| 2. | Select randomly generated ending from the stored list. (ing. end. axe. gzx or goh) |
| 3. | Add the partial student name with the randomly generated ending from the stored list |
| **Output** | |
| 1. | Display the partial student's name combined with the randomly generated ending |

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

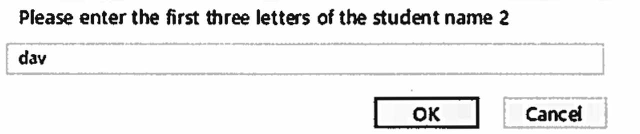
## Task 2b



```
//Declare variables
global studentname, ending, username, loopnumber, randomnumber,

on mouseUp
   //Declare subroutines
initialise
makeusername
end mouseUp

on initialise
put "" into studentname
put "" into username
put ing into ending[1]
put end into ending[2]
put axe into ending[3]
put gex into ending[4]
put goh into ending[5]

//Ask user for number of usernames they want to be generated
ask"How many students usernames do you want to make?"
if the result=cancel then exit to top
put it into loopnumber
end initialise

on makeusername
   //Ask user to enter student names
   repeat with counter = 1 to (loopnumber)
      ask "Please enter the first three letters of the student name" && counter
      if the result=cancel then exit to top
      put it into studentname[counter]

      //Ensure the user only enters the first three letters of the name
      repeat until len(studentname[counter]) = 3
         ask "Error, Please only enter the first three letters."
         if the result = cancel then exit to top
         put it into studentname[counter]
         end repeat
         put random(5) into randomnumber

         //Display generated username
         put "username " & counter& ":" && studentname[counter] & ending[randomnumber] & return after field "output1"

      end repeat
   end makeusername


```

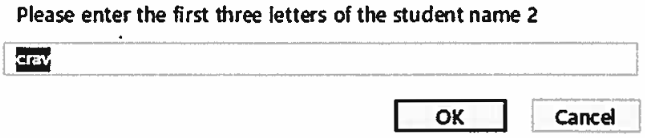


```
username 1: daving
username 2: cragex
```



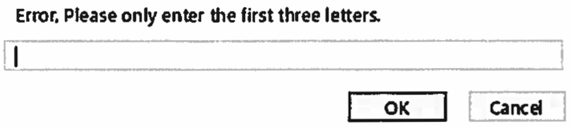
Please enter the first three letters of the student name 2

dav

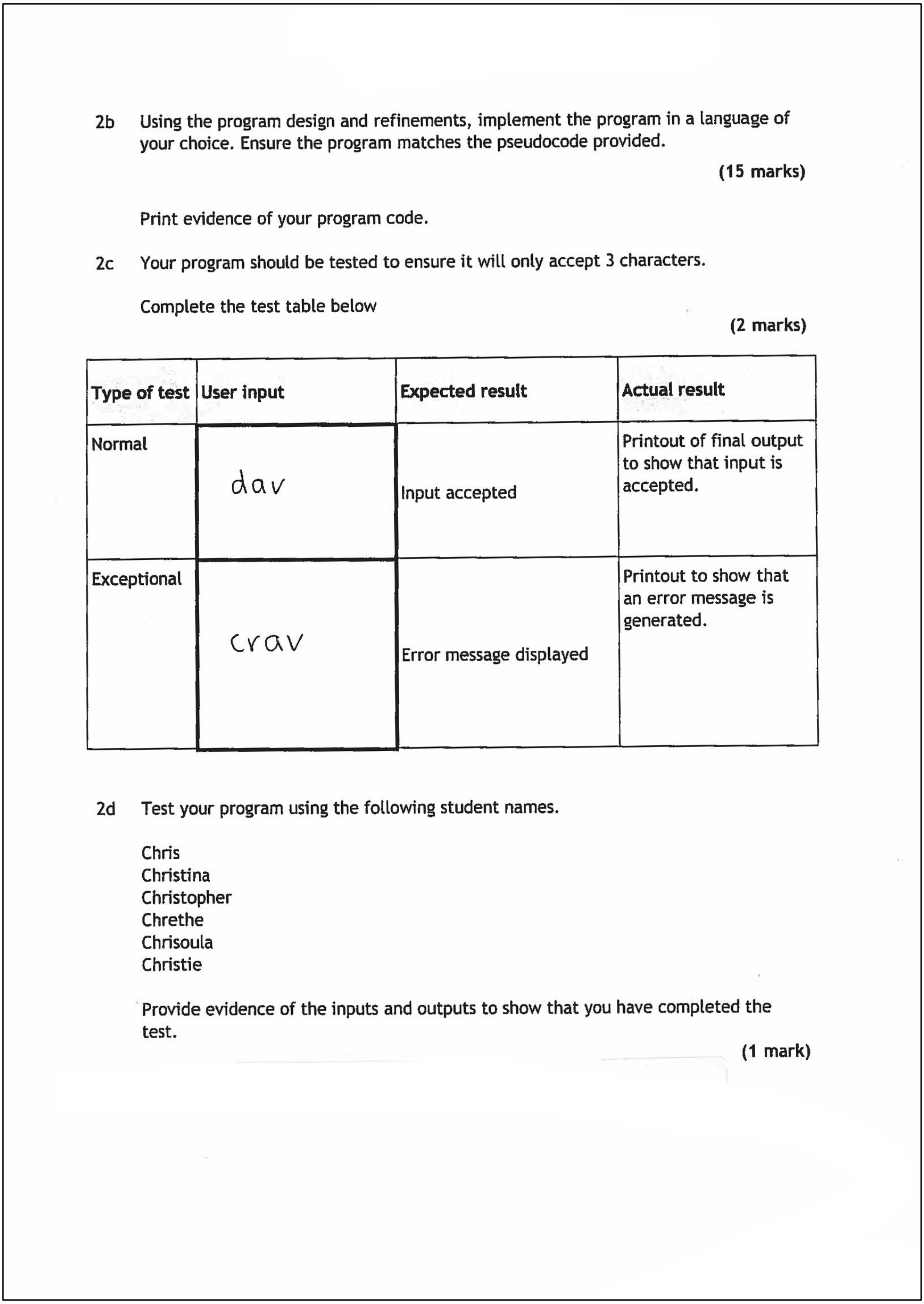


Please enter the first three letters of the student name 2

crav



Error, Please only enter the first three letters.



2b Using the program design and refinements, implement the program in a language of your choice. Ensure the program matches the pseudocode provided.

**(15 marks)**

Print evidence of your program code.

2c Your program should be tested to ensure it will only accept 3 characters.

Complete the test table below

**(2 marks)**

| **Type of test** | **User input** | **Expected result** | **Actual result** |
|---|---|---|---|
| Normal | dav | Input accepted | Printout of final output to show that input is accepted. |
| Exceptional | crav | Error message displayed | Printout to show that an error message is generated. |

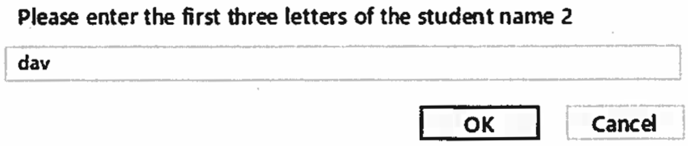
2d Test your program using the following student names.

Chris
Christina
Christopher
Chrethe
Chrisoula
Christie

Provide evidence of the inputs and outputs to show that you have completed the test.

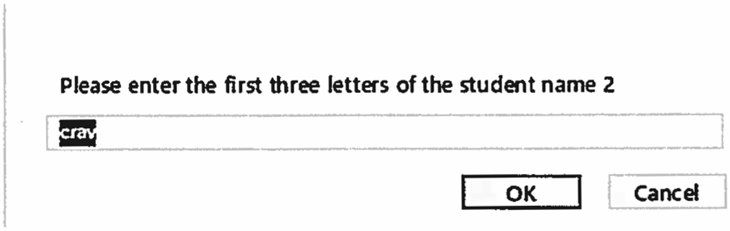
**(1 mark)**

Task 2c



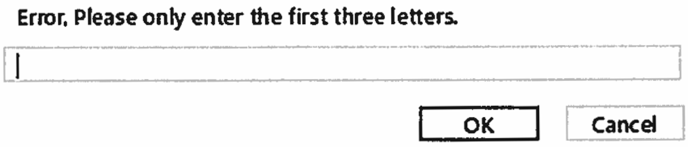
Please enter the first three letters of the student name 2

dav

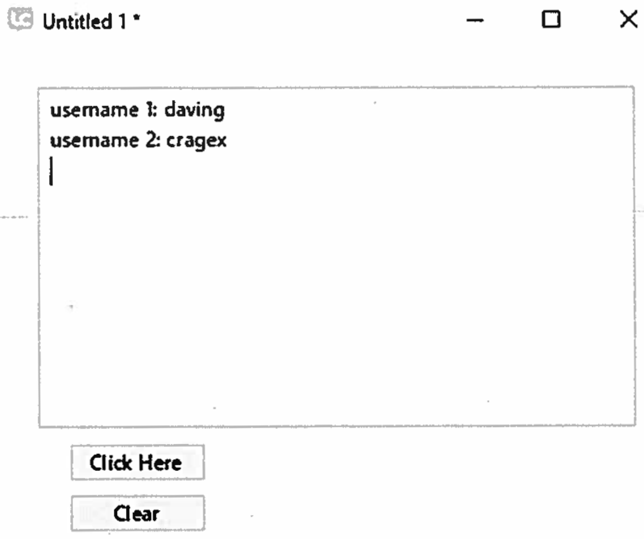


Please enter the first three letters of the student name 2

crav

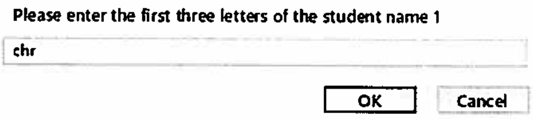


Error, Please only enter the first three letters.

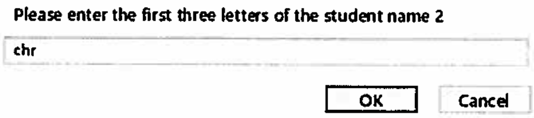


```
username 1: daving
username 2: cragex
```

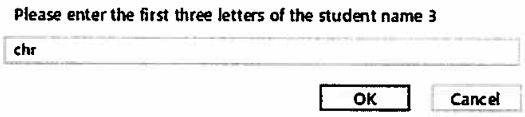
Task 2d


Please enter the first three letters of the student name 1

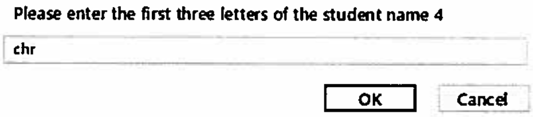
chr


Please enter the first three letters of the student name 2

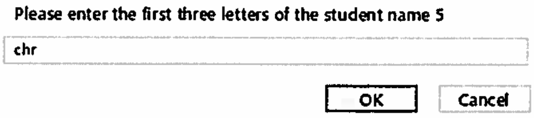
chr


Please enter the first three letters of the student name 3

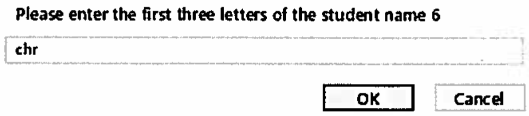
chr


Please enter the first three letters of the student name 4

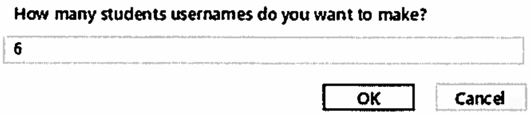
chr


Please enter the first three letters of the student name 5

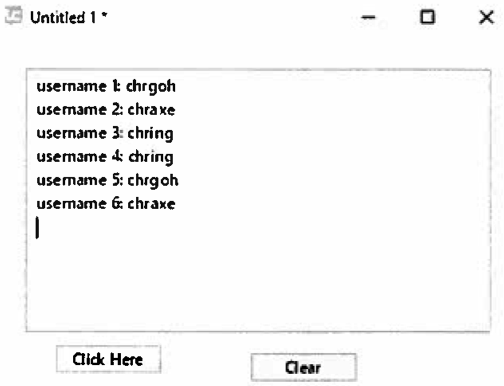
chr


Please enter the first three letters of the student name 6

chr


How many students usernames do you want to make?

6



```
username 1: chrgoh
username 2: chraxe
username 3: chring
username 4: chring
username 5: chrgoh
username 6: chraxe
```

**2e** **With reference to your code and testing, evaluate your own program by commenting on the following:**

**Efficient use of programming constructs in your code.**

**(1 mark)**

my use of loops was very effective as by using a loop, I didn't have to type out every part of code repeatedly. for example, the "repeat with counter = 1 to (loopnumber)" automatically repeats the following code the number of times the user has entered, therefore more efficient

**Robustness of your completed program**

**(1 mark)**

My program is partly robust as it crashes if you input text when it asks for 'how many student names do you want to make' but can withhold and display an error message when requirements aren't met when asked "Please enter first three letters of student name'.

**The readability of your code**

**(1 mark)**

I used internal commentary which makes the code much easier to read, as it explains what the code is doing

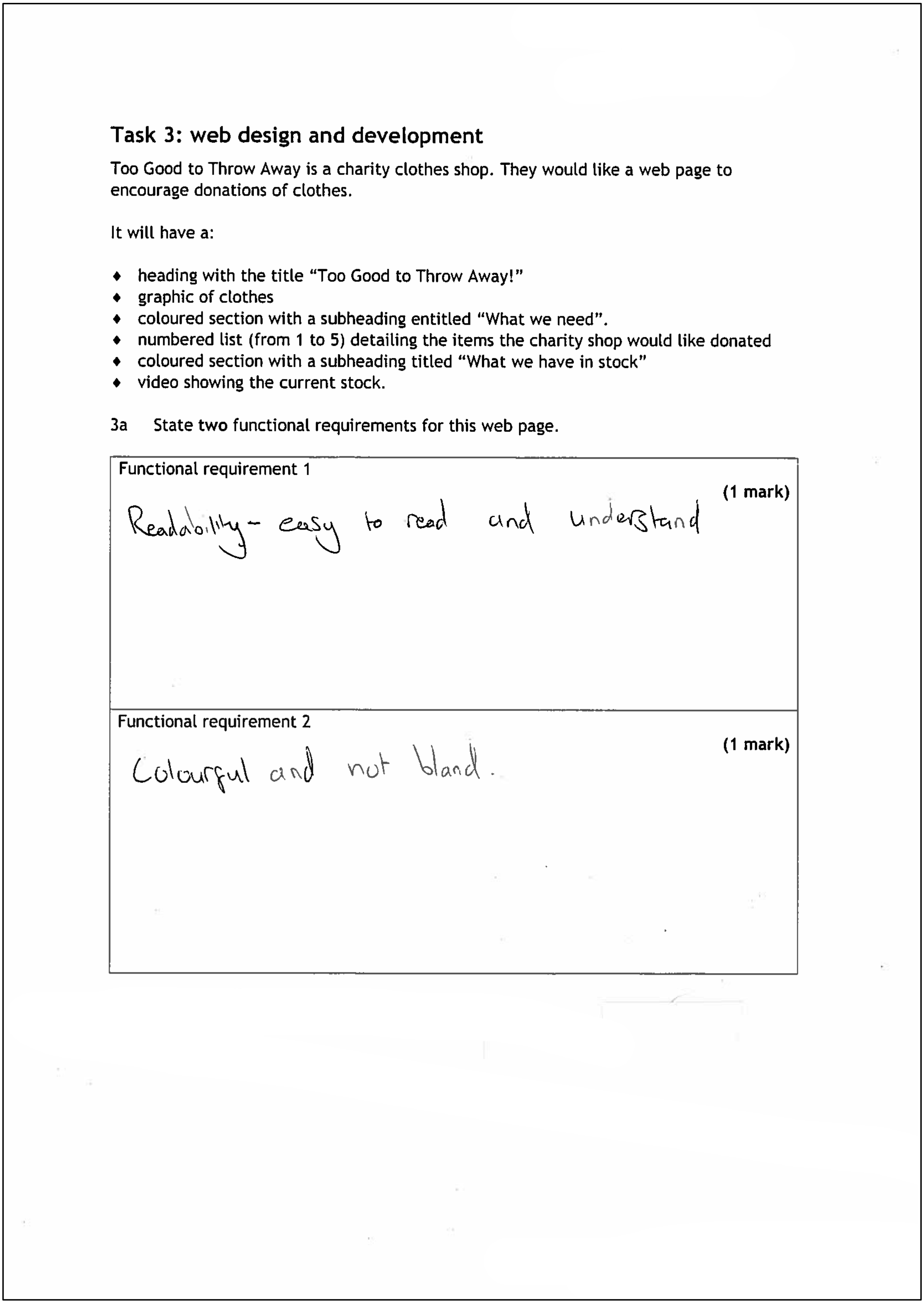
(example, "//Display generated username" tells the user that the following code is used to display usernames)

**Evaluate the fitness for purpose of the solution**

**(1 mark)**

My program is reasonably fit for purpose as it meets the requirements of generating and displaying usernames, only allowing 3 letters to be entered. However, it does not stop the issue of having many of the same username, making it less fit for purpose.

# Web design and development



## Task 3: web design and development

Too Good to Throw Away is a charity clothes shop. They would like a web page to encourage donations of clothes.

It will have a:

- heading with the title "Too Good to Throw Away!"
- graphic of clothes
- coloured section with a subheading entitled "What we need".
- numbered list (from 1 to 5) detailing the items the charity shop would like donated
- coloured section with a subheading titled "What we have in stock"
- video showing the current stock.

3a State two functional requirements for this web page.

Functional requirement 1

**(1 mark)**

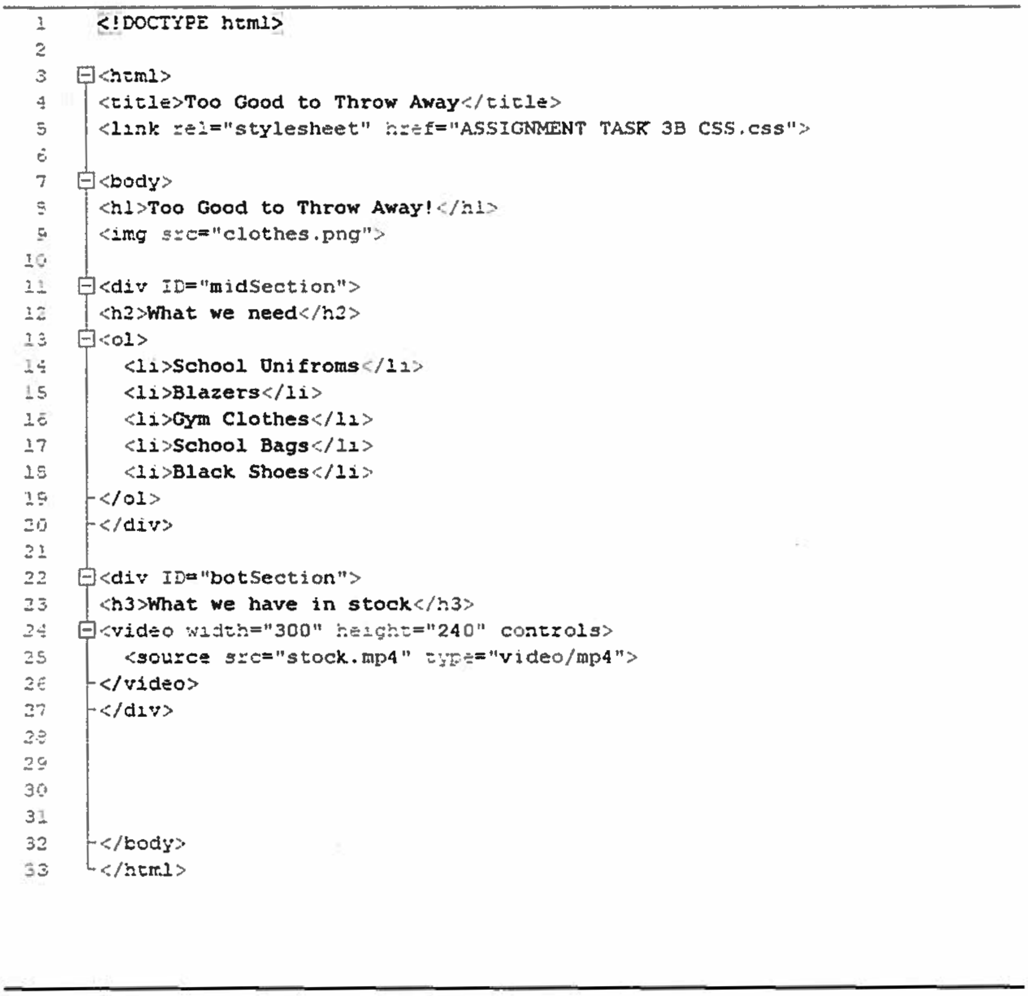
Readability - easy to read and understand

Functional requirement 2

**(1 mark)**

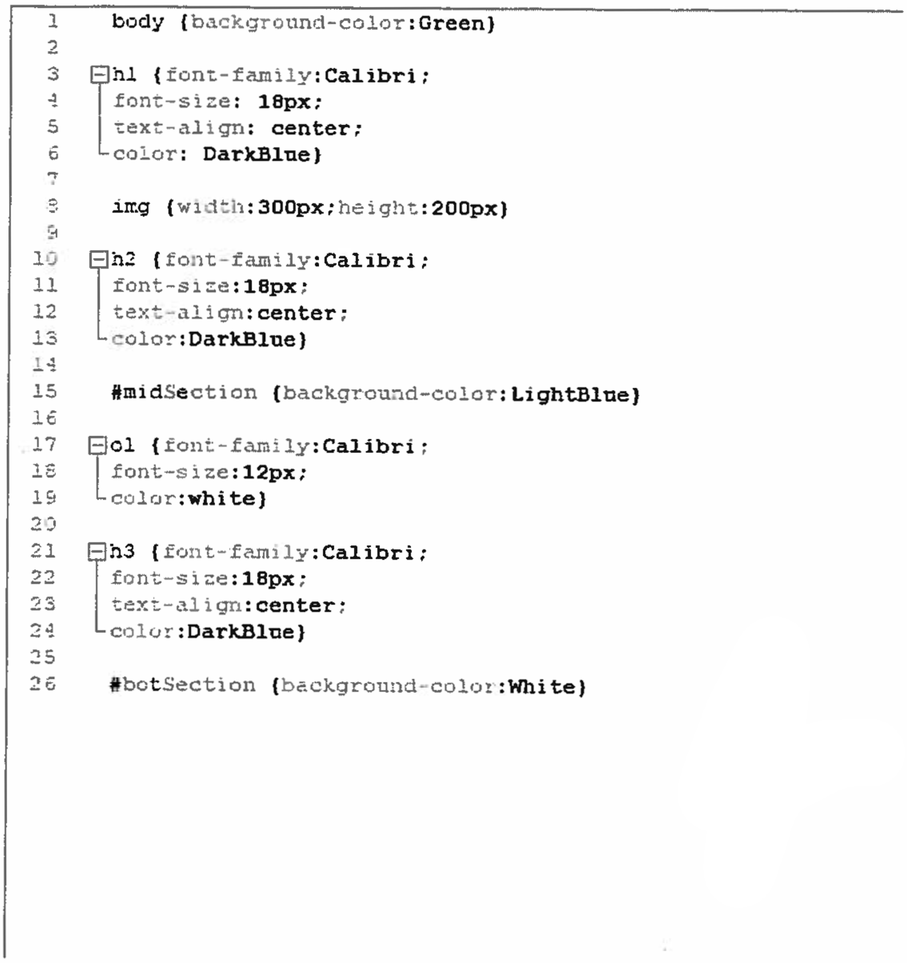
Colourful and not bland.

## HTML



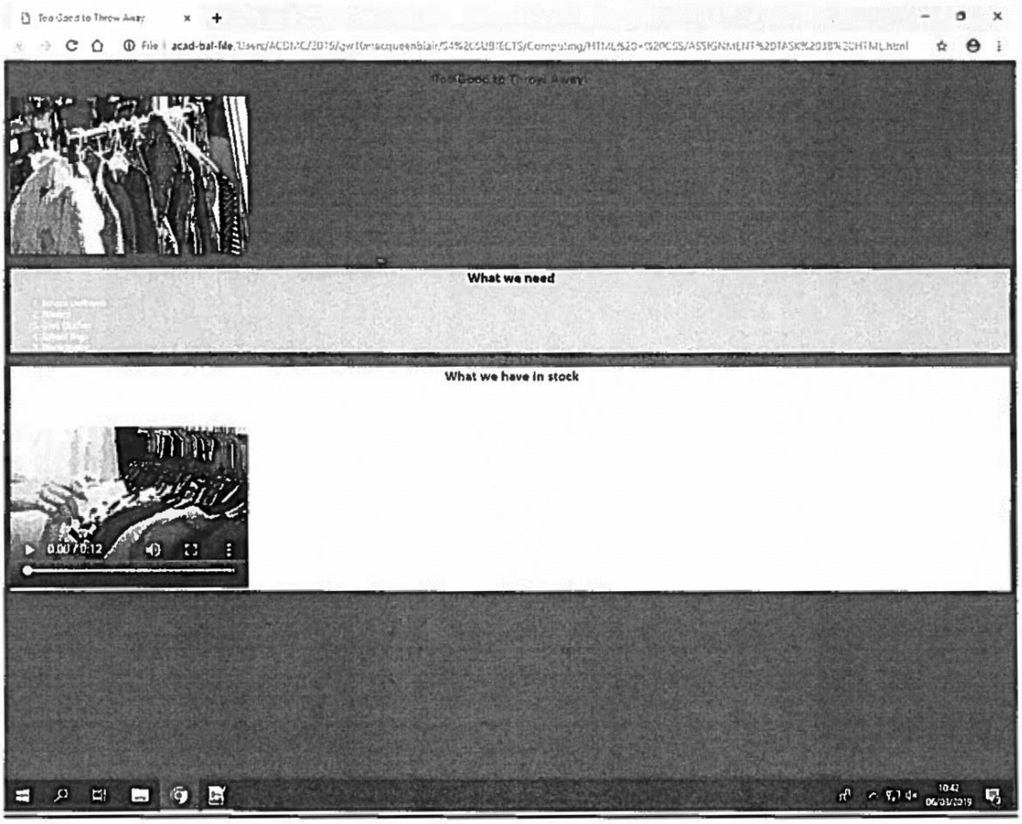
```
<!DOCTYPE html>

<html>
<title>Too Good to Throw Away</title>
<link rel="stylesheet" href="ASSIGNMENT TASK 3B CSS.css">

<body>
<h1>Too Good to Throw Away!</h1>
<img src="clothes.png">

<div ID="midSection">
<h2>What we need</h2>
<ol>
  <li>School Unifroms</li>
  <li>Blazers</li>
  <li>Gym Clothes</li>
  <li>School Bags</li>
  <li>Black Shoes</li>
</ol>
</div>

<div ID="botSection">
<h3>What we have in stock</h3>
<video width="300" height="240" controls>
  <source src="stock.mp4" type="video/mp4">
</video>
</div>




</body>
</html>
```

## CSS



```
body {background-color:Green}

h1 {font-family:Calibri;
font-size: 18px;
text-align: center;
color: DarkBlue}

img {width:300px;height:200px}

h2 {font-family:Calibri;
font-size:18px;
text-align:center;
color:DarkBlue}

#midSection {background-color:LightBlue}

ol {font-family:Calibri;
font-size:12px;
color:white}

h3 {font-family:Calibri;
font-size:18px;
text-align:center;
color:DarkBlue}

#botSection {background-color:White}
```

## WEB PAGE VIEWED IN BROWSER



Too Good to Throw Away!

**What we need**

**What we have in stock**

3c Describe two tests that could be performed on this web page.

**Test 1**

**(1 mark)**

To See if the video works

**Test 2**

**(1 mark)**

Seeing if the CSS In linked with the html

3d With reference to your solution, evaluate your web page by commenting on the following:

**Fitness for purpose**

**(1 mark)**

The web page met the Specification points by using div tags, an ordered list, correct fonts ~~and~~, Colours and Sizes, Images and videos.