National 5 Computing Science Assignment Assessment task

This document provides information for teachers and lecturers about the coursework component of this course in terms of the skills, knowledge and understanding that are assessed. It must be read in conjunction with the course specification.

Valid for session 2019-20 only.

Contents

Introduction

This document contains instructions for teachers and lecturers, and instructions for candidates for the National 5 Computing Science assignment. You must read it in conjunction with the course specification.

This assignment has 50 marks out of a total of 160 marks available for the course assessment.

This is one of two course assessment components. The other component is a question paper.

Instructions for teachers and lecturers

This assessment applies to the assignment for National 5 Computing Science for the academic session 2019-20.

The task is valid for 2019-20 only. Once complete, you must send the assignment responses to SQA to be marked.

You must conduct the assignment under a high degree of supervision and control. This means:

♦ candidates must be supervised throughout the session(s)
♦ candidates must not have access to email or mobile phones
♦ candidates must complete their work independently — no group work is permitted
♦ candidates must not interact with each other
♦ with no interruption for targeted learning and teaching
♦ in a classroom environment

Time

Candidates have 8 hours to carry out the assignment, starting at an appropriate point in the course, after all content has been delivered. It is not anticipated that this is a continuous 8-hour session, although it can be, but conducted over several shorter sessions. This is at your discretion.

You have a responsibility to manage candidates' work, distributing it at the beginning and collecting it in at the end of each session, and storing it securely in between. This activity does not count towards the total time permitted for candidates to complete the assignment.

Candidates are prompted to print their work at appropriate stages of the tasks. They can print on an ongoing basis or save their work and print it later. Whatever approach they take, time for printing is not part of the 8 hours permitted for the assignment.

Resources

Each candidate must have access to a computer system with a high-level (textual) programming language, database application and software that can create, edit and run SQL, HTML and CSS.

This is an open-book assessment. Candidates can access resources such as programming manuals, class notes, textbooks and programs they have written throughout the course. These may be online resources.

You must not create learning and teaching tasks that make use of constructs required in the assessment task, **with the specific purpose of developing a solution that candidates can access during the assignment.**

There may be instances where restriction of network use is prohibited (for example, a local authority-managed network with specific limitations). However, it remains your professional responsibility to make every effort to meet the assessment conditions.

Reasonable assistance

The assignment consists of three independent tasks. They are designed in a way that does not require you to provide support to candidates, other than to ensure that they have access to the necessary resources. Candidates can complete the tasks in any order.

Once the assignment is complete, you must not return it to the candidate for further work to improve their mark. You must not provide feedback to candidates or offer an opinion on the perceived quality or completeness of the assignment response, at any stage.

You can provide reasonable assistance to support candidates with the following aspects of their assignments:

♦ printing, collating and labelling their evidence to ensure it is in the format specified by SQA
♦ ensuring candidates have all the materials and equipment required to complete the assignment — this includes any files provided by SQA
♦ ensuring candidates understand the conditions of assessment and any administrative arrangements around the submission and storage of evidence, and the provision of files
♦ technical support

Evidence

All candidate evidence (whether created manually or electronically) must be submitted to SQA in a paper-based format. The evidence checklist details all evidence to be gathered. You can use it to ensure you submit all evidence to SQA.

You should advise candidates that evidence, especially code, must be clear and legible. This is particularly important when pasting screenshots into a document.

There is no need for evidence to be printed single sided or in colour.

Alteration or adaptation

The tasks are in PDF and Word formats. Each task is available as a separate file from the secure site. Word files allow candidates to word process their responses to parts of the task.

You must not adapt the assignment in any way that changes the instructions to the candidate and/or the nature and content of the tasks. However, you can make changes to font size, type and colour and to the size of diagrams for candidates with different assessment needs, for example, visual impairment.

If you are concerned that any particular adaptation changes the nature and/or the content of the task, please contact our Assessment Arrangements team for advice as soon possible at aarequests@sqa.org.uk.

Submission

Each page for submission has the number of the assignment task that it refers to, for example 1a, and contains space for candidates to complete their name and candidate number. Any other pages submitted, for example, prints of program listings or screenshots, must have this information added to them.

Specific instructions for teachers and lecturers: 2019-20

You must follow these specific instructions and ensure that candidates are aware of what you will give them at each stage in the assessment.

Print each task on single-sided paper, where applicable:

♦ this allows candidates to refer to information on other pages
♦ this helps you manage tasks that are split into more than one part

Task 1 — part A requires candidates to analyse a database. They must submit their evidence for this part of the task before starting part B.

Task 1 — part B is a separate section. This ensures that candidates do not access part A and change their responses. A Microsoft Access file (anytimeFlowers) is provided for candidates to use in part B. If your centre uses a different database management system, you can create the relational database for part B using the CSV files provided:

♦ customer.csv
♦ flowerOrder.csv

If using the CSV files you should set up all tables, fields and validation shown in the data dictionaries below. Some validation has been intentionally left blank to be completed as part of the task. Referential integrity should also be enforced.

Entity: Customer					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
customerID	PK	number		Y	
forename		text	40	Y	
surname		text	50	Y	
address		text	100	N	
telephoneNo		text	11	N	

Entity: FlowerOrder					
Attribute name	Key	Type	Size	Required	Validation
orderID	PK	text	10	Y	
dateDue		date		Y	
price		number		Y	
flowerType		text	8	Y	Restricted choice: rose, lily, tulip, daffodil
bunchSize		text	6	Y	Restricted choice: small, medium, large
chocolates		Boolean		Y	
message		text	200	N	
customerID	FK	number		Y	Existing customerID from Customer table

Task 2 — part A requires candidates to refine one process identified in the flowchart design. They must submit their evidence for this part of the task to you before starting part B which contains a possible solution to part A.

Task 2 — part B is a separate section. This ensures that candidates are not able to access part A and change their responses.

Task 3 — the following compressed file has been provided (winterwoollies.zip). When unpacked this contains the CSS, HTML and images candidates need to complete this task. These files should remain in the folder and should not be renamed.

Candidates **do not** need to print completed web pages in colour.

Instructions for candidates

This assessment applies to the assignment for National 5 Computing Science.

This assignment has 50 marks out of a total of 160 marks available for the course assessment.

It assesses the following skills, knowledge and understanding:

♦ applying aspects of computational thinking across a range of contexts
♦ analysing problems within computing science across a range of contemporary contexts
♦ designing, implementing, testing and evaluating digital solutions (including computer programs) to problems across a range of contemporary contexts
♦ demonstrating skills in computer programming
♦ applying computing science concepts and techniques to create solutions across a range of contexts

Your teacher or lecturer will let you know if there are any specific conditions for doing this assessment.

In this assessment, you have to complete three short practical tasks. You may complete the tasks in any order.

Advice on how to plan your time

You have 8 hours to complete the assignment. Marks are allocated as follows:

♦ Task 1 — database design and development   12 marks   (24% of total)
♦ Task 2 — software design and development   25 marks   (50% of total)
♦ Task 3 — web design and development   13 marks   (26% of total)

You can use this split as a guide when planning your time for each of the three tasks.

Advice on gathering evidence

As you complete each task, you must gather evidence as instructed in each task.

Your evidence, especially code, must be clear and legible. This is particularly important when you paste screenshots into a document.

Use the evidence checklist provided to make sure you submit everything necessary at the end of the assignment. Ensure your name and candidate number is included on all your evidence.

Evidence may take the form of printouts of code/screenshots/typed answers, hand-written answers or drawings of diagrams/designs.

Advice on assistance

This is an open-book assessment. This means that you can use:

♦ any classroom resource as a form of reference (for example programming manuals, class notes, and textbooks) — these may be online resources
♦ any files you have previously created throughout the course

The tasks are designed so you can complete them independently, without any support from your teacher or lecturer. This means that you:

♦ cannot ask how to complete any of the tasks
♦ cannot access any assignment files outside the classroom

Computing Science assessment task: evidence checklist

Task 1	Evidence	
Part A		
1a	Completed task 1 sheet showing analysis of database inputs	☐
Part B		
1b	Printout or screenshot showing correct validation has been set up on the two identified fields in the database	☐
1c (i)	SQL statement to change customer's order	☐
	Printout of the updated FlowerOrder table	☐
1c (ii)	SQL statement to add the new customer	☐
	Printout of the updated Customer table	☐
1d	Completed task 1 sheet showing two reasons why the SQL statement failed	☐
Task 2	**Evidence**	
Part A		
2a	Completed task 2 sheet showing design for 'calculate final bill using random value'	☐
Part B		
2b	Printout of your program code	☐
2c (i)	Completed task 2 sheet showing possible values for final bill	☐
	Printout evidence of one test run showing inputs and outputs	☐
2c (ii)	Completed task 2 sheet showing completed test table	☐
	Printout evidence of test run of exceptional test data	☐
2d	Completed task 2 sheet showing completed evaluation	☐
Task 3	**Evidence**	
3a	Completed task 3 sheet showing the end-user and functional requirements	☐
3b	Completed task 3 sheet showing wireframe design for 'Orders' page	☐
3c and 3d	Printouts of HTML and CSS code: ♦ orders.html ♦ styles.css	☐
3e	Completed task 3 sheet showing completed evaluation	☐

Please follow the steps below before handing your evidence to your teacher or lecturer:

♦ Check you have completed all parts of tasks 1, 2 and 3.
♦ Label any printouts and/or screenshots with the task number (for example 1c, 2a).
♦ Clearly display your name and candidate number on each printout.

Task 1: database design and development (part A)

Anytime Flowers is a florist that tailor-makes bunches of flowers for customers.

When a customer comes into the shop, an employee notes down the customer's details, their order information and the price of the order. They give each order an order code.

Customers select one type of flower (rose, lily, tulip or daffodil), the size of the bunch of flowers (small, medium or large) and whether or not they would like chocolates with the flowers. They also select a date for collecting their order. Customers can include a message with their flowers.

1a   Anytime Flowers wants to create a database to store customer and order details.

Complete the order details in the analysis of inputs table below:

(2 marks)

Customer details:	Order details:
Customer ID Name Address Telephone number	

- 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.

Candidate name_____ Candidate number_____

Task 1: database design and development (part B)

1b   Your teacher or lecturer will provide you with a database file containing two linked tables.

Using the data dictionary below, complete the relational database by:

♦   identifying two fields where the validation shown below has yet to be applied
♦   adding the validation to the two identified fields

(2 marks)

Entity: Customer					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
customerID	PK	number		Y	
forename		text	40	Y	
surname		text	50	Y	
address		text	100	N	
telephoneNo		text	11	N	Length = 11

Entity: FlowerOrder					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
orderID	PK	text	10	Y	
dateDue		date		Y	
price		number		Y	Range: >= 5.00 and <= 50.00
flowerType		text	8	Y	Restricted choice: rose, lily, tulip, daffodil
bunchSize		text	6	Y	Restricted choice: small, medium, large
chocolates		Boolean		Y	
message		text	200	N	
customerID	FK	number		Y	Existing customerID from Customer table

Print evidence to show that you have added the validation to the database to match the data dictionary requirements.

1c (i) A customer would like to change their order from 'rose' to 'tulip'. The price of the order will change from £34 to £17. The orderID is CHQ3848.

Implement **one** SQL statement that will make the required changes to the order.

(4 marks)

Print evidence of the SQL statement and the FlowerOrder table, clearly showing that the changes have been implemented.

(ii) A new customer provides their name and telephone number.

Implement an SQL statement that will add their details to the database.

Name:	Richard Glass
Telephone number:	07654029336

(2 marks)

Assign them customerID — 2986.

Print evidence of the SQL statement and the Customer table, clearly showing that the changes have been implemented.

1d   Anytime Flowers wants to find the names of all customers who had placed orders for the smallest bunch of flowers.

The following incorrect SQL statement is written.

```
SELECT customerName
FROM Customer, FlowerOrder
WHERE size = "smallest"
AND Customer.customerID = FlowerOrder.customerID;
```

Test this SQL statement.

State two reasons why this SQL statement failed.

(2 marks)

Reason 1
Reason 2

Task 2: software design and development (part A)

A new coffee shop is organising an event for its opening day. At this event, a lucky-dip promotion will be available, where customers can win a discount off their bill.

Below is the analysis and design for a program to calculate customers' bills:

Program analysis
A program is required to calculate a customer's bill. The user will enter the number of items on the bill and then enter the item type for each item (coffee, tea or biscuit). The program will calculate the bill. The bill can then be reduced by using a random value from 1 to 10:

♦ random value = 1   the customer pays nothing
♦ random value = 2 to 6   the customer pays half the bill
♦ random value = 7 to 10   the customer pays the full bill

Assumptions
♦ any number of items can be entered by the user

Inputs
♦ the number of items on the bill
♦ the item type for each item on the bill
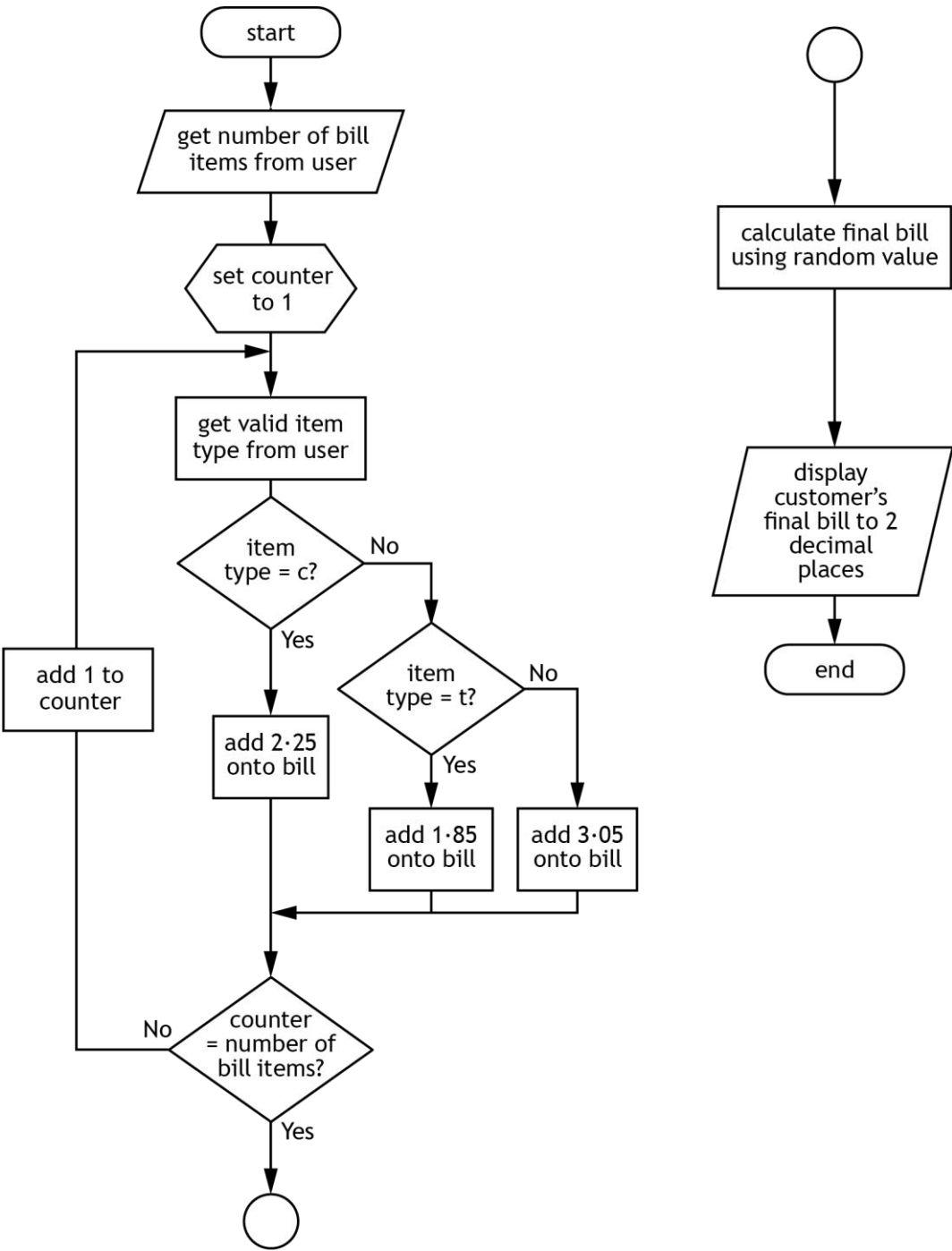 — c = coffee
  — t = tea
  — b = biscuit

Processes
♦ generate a random value between 1 and 10
♦ calculate the total cost of the items on the bill where:
  — coffee = £2.25
  — tea = £1.85
  — biscuit = £3.05
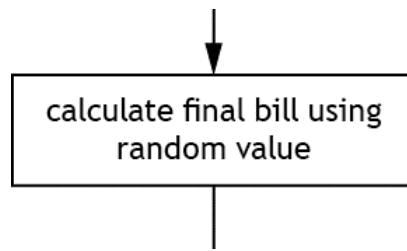♦ use the random value to calculate the final bill

Outputs
♦ the random value
♦ the cost of the final bill

Program design (flowchart)

start

get number of bill
items from user

set counter
to 1

get valid item
type from user

item type = c? — No → item type = t? — No → add 3·05 onto bill

Yes ↓ (item type = c)
add 2·25 onto bill

Yes (item type = t) → add 1·85 onto bill

add 1 to counter

counter = number of bill items? — No → (add 1 to counter)

Yes ↓

(end of loop)

calculate final bill using random value

display customer's final bill to 2 decimal places

end

2a The flowchart contains the following process:

Using the information provided in the program analysis, expand the design to show how this process could be carried out. You can use a flowchart, structure diagram or pseudocode design.
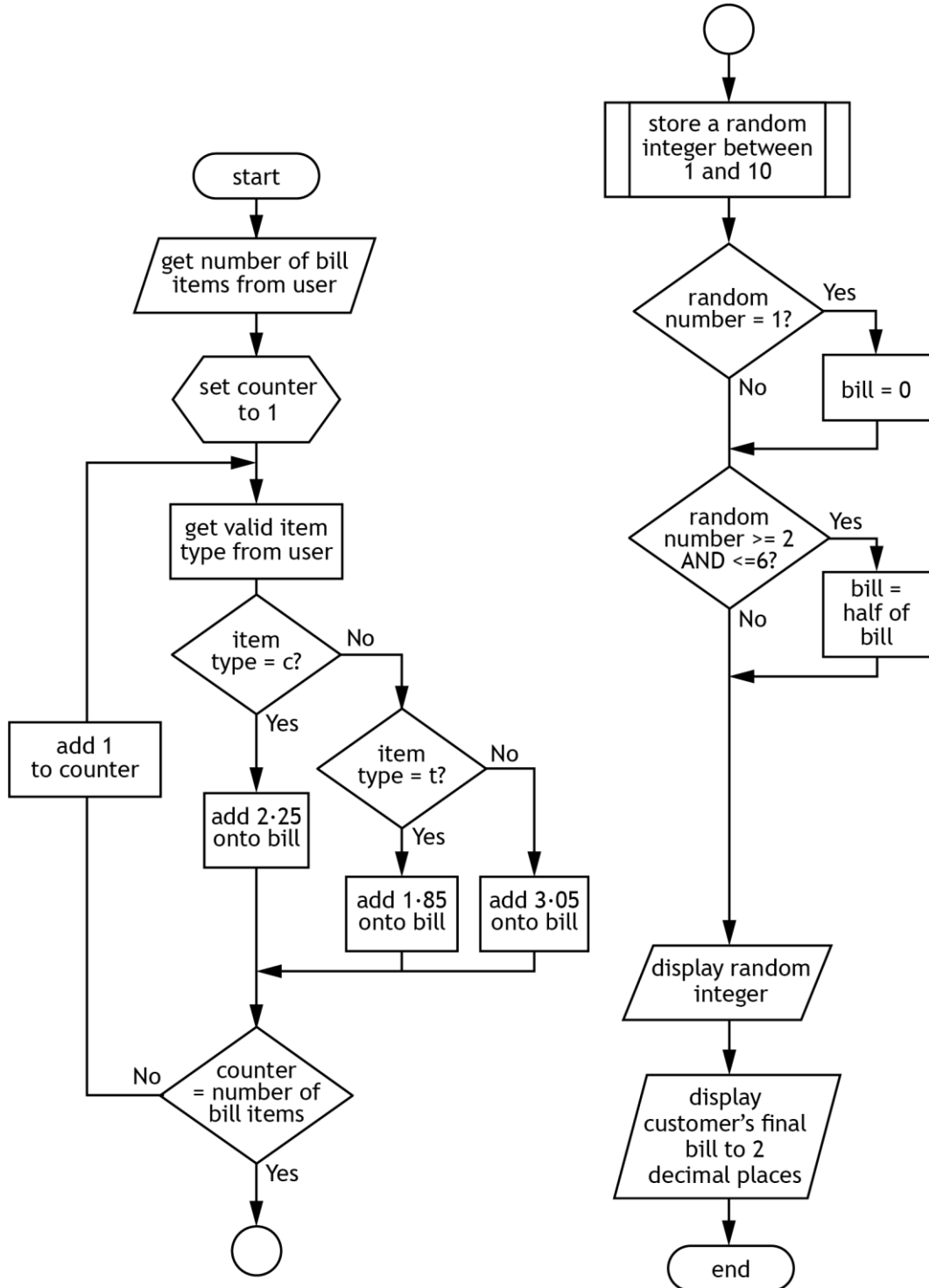
(3 marks)

- 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.

Candidate name_____ Candidate number_____

Program design (completed flowchart)

2b   Using the program analysis and the design, implement the program in a language of your choice.

Ensure the program matches the flowchart provided on page 17.

(15 marks)

Print evidence of your program code.

2c (i) Your program should be tested to ensure it produces one of three different random outputs.

Use the following data to do this:

Number of items: 4
Item 1: coffee
Item 2: tea
Item 3: tea
Item 4: biscuit

State the possible values (outputs) for the final bill produced from this test data.

Run your program to show that it produces one of these three outputs.

Print evidence of the test run showing inputs and outputs.

(2 marks)

(ii)   Complete the test table below to check the validation for the item type.

Type of test	Test data	
Normal	Number of items – 3	c, t, b
Exceptional	Number of items - 1	

Run your program to show the result of the exceptional test data.

(1 mark)

Print evidence of the test run.

Candidate name_____ Candidate number_____

2d   With reference to your code, evaluate your program by commenting on the following:

Efficiency of your program code	**(2 marks)**

Robustness of your completed program	**(1 mark)**

Readability of your code	**(1 mark)**

Candidate name_____ Candidate number_____

Task 3: web design and development

Winter Woollies is a group of home knitters who makes and sells woollen hats, scarves and gloves.

It wants to create a website with the following content:

♦ the title 'Winter Woollies'
♦ a short statement about the group
♦ separate pages for each category of product: hats, scarves and gloves
♦ photographs of every product
♦ a video of one of the members of the group knitting
♦ an 'Orders' page with contact details and a link to an external online payment website

3a State **one** end-user and **two** functional requirements for the website.

End-user requirement	**(1 mark)**

Functional requirement 1	**(1 mark)**

Functional requirement 2	**(1 mark)**

Candidate name_____ Candidate number_____

3b   Your teacher or lecturer will provide you with a copy of the unfinished website.

Open this and look carefully at:

♦   the layout of each page
♦   the content of each page
♦   the navigation within the website

The 'Orders' page currently has no content. When complete, it should match the layout of the other pages and include the following content:

♦   a message stating that most orders are delivered within four weeks
♦   a message stating that orders can be paid using the external website www.payfriend.com
♦   a message stating that orders can be requested by sending an email to winter.woollies@homemail.com
♦   an image (200x100 pixels) of a parcel ready for posting
♦   a hyperlink back to the home page

Complete the wireframe on the following page, showing how you would lay out the page content for the 'Orders' page in a consistent manner to the rest of the website.

(2 marks)

Winter Woolies - Orders
`_` `⊡` `✕`

Group Name
Group Statement

500 x 60
banner.jpg

Candidate name_____ Candidate number_____

3c   Open the orders.html and styles.css files in order to edit them.

        Implement your design of the 'Orders' page web page using HTML, including all of
        the content and any hyperlinks required.

        The file 'parcel.jpg' has been provided within the website files.

        (4 marks)

3d   Feedback received from user testing of the website highlights that:

        ♦   the website would look better if the three coloured sections on each page were
            all the same colour
        ♦   the text in the statement 'Home knitting delivered to your door' should be the
            same font and size as the 'Winter Woollies' heading above it
        ♦   the link to www.payfriend.com needs to stand out more than the other links —
            the style of the text in the link should be white in colour and font size 12

        Edit the orders.html and styles.css files to implement these changes.

        (3 marks)

        Print evidence of your code from these edited files:

        ♦   orders.html
        ♦   styles.css

3e   Evaluate the website in terms of fitness for purpose

        (1 mark)

Candidate name_____ Candidate number_____

Administrative information

Published: January 2020 (version 1.0)

History of changes

Version	Description of change	Date

Security and confidentiality

This document can be used by SQA approved centres for the assessment of National Courses and not for any other purpose.

This document may only be downloaded from SQA's designated secure website by authorised personnel.

© Scottish Qualifications Authority 2020