



Fundamentals of Computing and Digital Literacy

Sample

Assignment title: Develop a Wiki

Marking Scheme

This Marking Scheme has been prepared as a guide only to markers. This is not a set of model answers, nor is the Marking Scheme exclusive, for there will frequently be alternative responses which will provide a valid answer. Unless a question specifies that an answer be produced in a particular form, then an answer that is correct, factually or in practical terms, must be given the available marks.

If there is doubt as to the correctness of an answer the relevant NCC Education materials and associated module textbook should be the first authority.

Throughout the question, please credit any valid alternative point.

Notice to Markers

Where markers award half marks in any part of a question they should ensure that the total mark recorded for a task is rounded up to a whole mark.

Marking Guidance

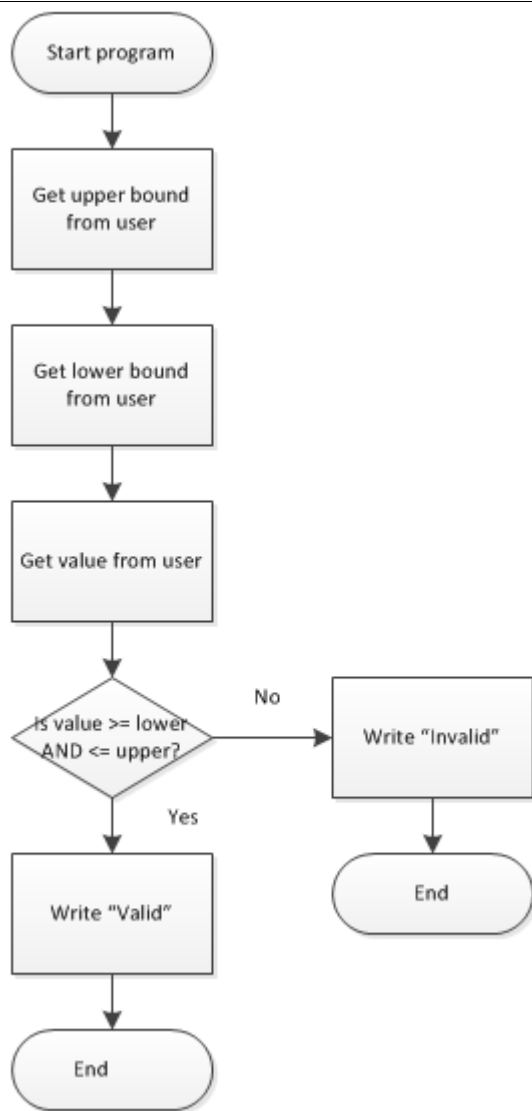
Task	Guide	Maximum Marks
1	<p>a) Award 5 marks for successfully registering for a Wiki. This will either be successful or unsuccessful, so either 5 marks or 0 marks will be awarded.</p> <p>b) Award 5 marks for setting up a new Wiki with the correct name. Deduct 2 marks if the name is incorrect.</p> <p>c) Award 1 mark for creating the page, 1 mark for the correct title, and 3 marks for suitable content.</p> <p>d) Award 1 mark for creating the page. Award 1 mark for the correct sub-heading. Award 2 marks for inserting the image into the wiki page. Award 1 mark for adding the correct caption to the image.</p> <p>e) A possible solution in pseudocode is given below:</p> <pre> COUNT = 0 REPEAT MOVE ADD 1 TO COUNT UNTIL COUNT >= 4 TURN LEFT COUNT = 0 OUTER COUNT = 0 REPEAT REPEAT MOVE ADD 1 TO COUNT UNTIL COUNT >= 4 TURN RIGHT ADD 1 TO OUTER COUNT UNTIL OUTER COUNT >= 2 COUNT = 0 REPEAT MOVE ADD 1 TO COUNT UNTIL COUNT >= 6 TURN LEFT COUNT = 0 REPEAT MOVE ADD 1 TO COUNT UNTIL COUNT >= 3 TURN LEFT COUNT = 0 REPEAT MOVE ADD 1 TO COUNT </pre>	<p>5</p> <p>5</p> <p>5</p> <p>5</p> <p>23</p>

Task	Guide	Maximum Marks
	<p>UNTIL COUNT >= 3 TURN RIGHT REPEAT MOVE ADD 1 TO COUNT UNTIL COUNT >= 3</p> <p>Award 5 marks for correctly navigating the path, deducting 1 mark for each error (e.g. going 1 step too far) up to a maximum of 5 marks deducted.</p> <p>Award 5 marks for use of loops when moving 2 or more steps at a time, deducting 1 mark for each time a loop could be used but is not up to a maximum of 5 marks deducted.</p> <p>Award 2 marks for use of a counter to track number of steps.</p> <p>Award 2 marks for incrementing the counter at the correct points.</p> <p>Award 2 marks for resetting the counter at the correct points.</p> <p>Award 5 marks for use of a nested loop for stages 2 and 3 of the map, using an outer and inner loop counter, deducting 1 mark for each error in the logic for this.</p> <p>Award 2 marks for adding the pseudocode to the Wiki page with the correct subheading.</p> <p>f) A possible solution in pseudocode is given below:</p> <pre> WHILE AT_HOUSE IS TRUE WHILE PATH_AHEAD IS TRUE MOVE END WHILE WHILE PATH_AHEAD IS FALSE TURN RIGHT END WHILE END WHILE </pre> <p>Award 5 marks for correctly navigating the path.</p> <p>Award 5 marks for a loop which loops until the smiley face is at the house.</p> <p>Award 5 marks for a nested loop which keeps the face moving while there is path ahead.</p> <p>Award 5 marks for a nested loop which keeps the face turning right OR turning left until it finds path again.</p> <p>Award 2 marks for adding the pseudocode to the wiki page with the correct subheading.</p>	22

Task	Guide	Maximum Marks
	<p>g) Award 1 mark for the correct subheading. Award 1 mark for a suitable explanation of the flexibility, e.g. that the Path A solution can only solve that path whereas the General Solution can solve any path with no branches or breaks. Award 2 marks for each suitable disadvantage listed, up to a maximum of 4 marks. Possibilities include: long, inflexible, hard to debug, not generalizable. Award 2 marks for a suitable advantage of the general solution. Possibilities include: more flexible, works on a range of paths, easier to reuse. Award 2 marks for each suitable reason listed, up to a maximum of 4 marks. Possibilities include: easier to work out the specific solution than a general one, quicker to work out the specific solution, requires less 'deep thought' to work out the specific solution, can give the task to less experienced individuals if you only ask for a specific solution.</p> <p>h) A possible diagram is shown below:</p> <div data-bbox="284 936 1321 1592" data-label="Diagram"> <pre> graph TD CS[Character Stream] --> LA[Lexical Analysis (Lexer)] LA -- "Stream of Tokens" --> SA[Syntactic Analysis (Parser)] SA -- "Abstract Syntax Tree" --> SA_Sem[Semantic Analysis (Type Checker)] SA_Sem -- "Intermediate Code" --> CO[Code Optimization] CO -- "Optimized Intermediate Code" --> CG[Code Generation] CG --> MC[Machine Code] </pre> </div> <p>Award 1 mark for a suitable heading Award 1 mark for the correct input Award 2 marks for each correct stage (up to a maximum of 10 marks). Award 1 mark for the correct output. Award 2 marks for inserting the image or other type of file into the Wiki page. Award 1 mark for adding a suitable caption.</p>	<p>12</p> <p>16</p>

Task	Guide	Maximum Marks
	<p>i) Award 1 mark for suitable additional text on the Welcome page explaining that there are pages for Solving Problems with Algorithms and on The Compilation Process.</p> <p>Award 1 mark for a working link to the Solving Problems with Algorithms page from the Welcome page.</p> <p>Award 1 mark for a working link to The Compilation Process page from the Welcome page.</p> <p>Award 1 mark for a working link to the Path A image from the Welcome page.</p> <p>Award 1 mark for a working link to the compilation process image/ file from the Welcome page.</p> <p>Award 1 mark for a working link to the Welcome page from the Solving Problems with Algorithms page.</p> <p>Award 1 mark for a working link to the Welcome page from The Compilation Process page.</p>	7
	Total Maximum Mark	100

Task	Guide	Maximum Marks
2	<p>a) Award 5 marks for successfully registering for an account. This will either be successful or unsuccessful, so either 5 marks or 0 marks will be awarded.</p> <p>b) Award 5 marks for setting up a new Wiki with the correct name. Deduct 2 marks if the name is incorrect.</p> <p>c) Award 2 marks for creating the page. 1 mark for the correct title, and 2 marks for suitable content.</p> <p>d) Award 2 marks for creating the <i>Designing Computer Programs</i> page with a suitable heading. Award 2 marks for creating <i>The Importance of Design</i> page with a suitable heading. Award 2 marks for each suitable benefit given, up to a maximum of 4 marks. Possible benefits include: implementation independent (i.e. can be transferred into many different programming languages), easier to plan then when actually coding, easier to communicate the design to non-programmers this way. Award 1 mark for each suitable technique listed, up to a maximum of 2 marks. Possible answers include: pseudocode, flowcharts, UML diagrams, SSADM diagrams.</p> <p>e) Award 2 marks for creating the <i>Computer Program Design</i> page with a suitable heading. A possible solution in pseudocode is show below:</p> <pre> READ upper READ lower READ value IF value >= lower AND value <= upper Write "Valid" ELSE Write "Invalid" END IF </pre> <p>A possible solution in flowchart format is shown below:</p>	<p>5</p> <p>5</p> <p>5</p> <p>10</p> <p>10</p>



The pseudocode or flowchart does not need to follow these precise formats.

Award 1 mark for reading in the upper bound.

Award 1 mark for reading in the lower bound.

Award 1 mark for reading in the value to be checked.

Award 2 marks for the correct check (value is \geq lower and value \leq upper)

Award 1 mark for correct output statement if valid

Award 1 mark for correct output statement if invalid.

Award 1 mark for uploading the pseudocode or flowchart successfully to the correct Wiki page

f) A possible solution in VB.NET is given below:

10

```
Module Module1
```

```
Sub Main()
```

```
Dim upper As Integer
```

```
Dim lower As Integer
```

```
Dim value As Integer
```

```
Console.Out.WriteLine("Enter the upper bound")
```

```
upper = Console.ReadLine
```

```
Console.Out.WriteLine("Enter the lower bound")
```

```
lower = Console.ReadLine
```

```
Console.Out.WriteLine("Enter the value")
```

```
value = Console.ReadLine
```

```
If (value >= lower And value <= upper) Then
```

```
    Console.Out.WriteLine("Valid")
```

```
Else
```

```
    Console.Out.WriteLine("Invalid")
```

```
End If
```

```
Console.ReadKey()
```

```
End Sub
```

```
End Module
```

Award 1 mark for reading in the upper bound.

Award 1 mark for reading in the lower bound.

Award 1 mark for reading in the value to be checked.

Award 2 marks for the correct check (value is \geq lower and value \leq upper)

Award 1 mark for correct output statement if valid

Award 1 mark for correct output statement if invalid.

Award 1 mark for including a transcript of the code on the wiki page.

Award 2 marks for uploading the executable or program file to the wiki page.

g) A possible test plan is shown below:

10

Test no.	Test data	Expected result	Actual result
1	Upper = 12 Lower = 0 Value = 5	Valid	
2	Upper = 12 Lower = 0 Value = 0	Valid	
3	Upper = 12 Lower = 0 Value = 0	Valid	
4	Upper = 500 Lower = 100 Value = 99	Invalid	
5	Upper = 500 Lower = 100 Value = 100	Valid	
6	Upper = 500 Lower = 100 Value = 1001	Invalid	
7	Upper = 56 Lower = 2 Value = 22	Valid	
8	Upper = 56 Lower = 2 Value = "Twelve"	Invalid	

Award 1 mark for a suitable test case table design.
Award 1 mark for uploading or inserting the table into the Wiki page.
Award 1 mark for each suitable test case row including a suitable expected result, up to a maximum of 8 marks. Deduct 1 mark if no normal data tests are included. Deduct 1 mark if no boundary data tests are included. Deduct 1 mark if no invalid data tests are included.

h) Award 1 mark for each correctly documented test case result (i.e. where the actual result matches with what the candidates program code actually does) up to a maximum of 8 marks.
Award 4 marks for the discussion. The discussion should correctly identify any errors, or offer a valid explanation as to why no errors were found.

12

	<p>i) Award 1 mark for a working link to the <i>Designing Computer Programs</i> page from the <i>Welcome</i> page. Award 1 mark for a working link to the <i>Computer Program Design</i> page from the <i>Welcome</i> page. Award 1 mark for a working link to the <i>Writing Computer Programs</i> page from the <i>Welcome</i> page. Award 1 mark for a working link to the <i>Testing Computer Programs</i> page from the <i>Welcome</i> page. Add 1 mark for each link from these pages back to the <i>Welcome</i> page, up to a maximum of 4 marks.</p> <p>j) Award 1 mark for creating the <i>Accessible Hardware</i> page and giving it a suitable heading. Award 6 marks for the discussion of the impact of common physical impairments on computer use. Give 2 marks for each valid point. Possible suitable discussion points include the following:</p> <ul style="list-style-type: none"> • Standard input devices and output devices are not designed for use by visually impaired users. • Standard mice are very hard to control for users with steadiness issues such as motor neurone disease or cerebral palsy. • Alert sounds to indicate that you are doing something incorrect are not useful for hearing impaired users, and other feedback mechanisms via hardware (e.g. vibrations) are usually not included as standard. • Requirements to double click at default settings, hold down multiple keys at the same time, and so on, require a level of dexterity that many older users and those with steadiness issues such as motor neurone disease or cerebral palsy are often unable to achieve. <p>Award 8 marks for the discussion of accessible hardware device one and 8 marks for the discussion of accessible hardware device two. For each of the two devices, the marks should be broken down as follows: Award 1 mark for identification of a suitable device. Possible devices include, but are not limited to: braille keyboard, sip and puff mouse, braille writer for touchscreens, large print keyboards, foot mouse, joystick, on screen keyboard. Award 1 mark for including an image or link to a video of the device in action. Award 2 marks for an explanation of what the function of the device is. Award 2 marks for an explanation of how the device works. Award 2 marks for a discussion of what accessibility problem(s) the device solves.</p> <p>k) Award 1 mark for a working link to the <i>Accessible Hardware</i> page from the <i>Welcome</i> page. Award 1 mark for a working link from the <i>Welcome</i> page to the <i>Accessible Hardware</i> page</p>	<p>8</p> <p>23</p> <p>2</p>
	Total Maximum Mark	100

Learning Outcomes matrix

Task	Learning Outcomes assessed
1	LO2: Use principles of computational thinking to solve problems LO3: Be able to collaborate and communicate online LO6. Be able to develop, test and debug program code.
2	LO1: Understand the components of a computer system and its input devices. LO2: Use principles of computational thinking to solve problems. LO3: Be able to collaborate and communicate online LO6: Be able to develop, test and debug program code.