HANDOUT FOR CHAPTER 9

PROBLEM SOLVING AND DESIGN

Marking Scheme

Section B

2 1 mark for each error identified + suggested correction
Line 1 or Small = 0: this should read Small = 999
line 5 or IF…:this should read IF Num < Small THEN Small = Num
line 8 or UNTIL: this should read UNTIL Counter = 10 or
UNTIL Counter > = 10 or
UNTIL Counter > 9
line 7 or PRINT…: PRINT Small should come after the end of the repeat loop
or
line 8 or UNTIL: this should come before line 7 [4]

1

Total	Reject	Weight	Output
0	0		
1.8		1.8	
	1	26.0	
8.8		7.0	
20.1		11.3	
30.1		10.0	
32.6		2.5	
	2	25.2	
37.6		5.0	
57.4		19.8	
	3	29.3	
		–1	57.4, 3
(2 marks) (–1 for each error) (then follow though)	(1 mark)	1 mark)	(1 mark) (allow follow through) (from Total and Reject)

[5]

4 1 mark for each error identified + suggested correction

Line 1 or Large =9999: this should read Large = 0

Line 3 or WHILE: this should read WHILE Counter < 30

line 6 or IF: this should read IF Num > Large THEN Large = Num

line 7 or Counter =…: this should read Counter = Counter + 1 [4]

3 (a)

Trace table set 1

A	B	C	D	E	F	Total	Check	Output
5	2	4	3	1	5	38	5	Accept

---------- (1 mark)---------- (1 mark)------

Trace table set 2

A	B	C	D	E	F	Total	Check	Output
3	2	1	0	7	3	45	1	Reject

---------- (1 mark) ---------- (1 mark) ------

[4]

(b) – (modulo 11) check digit calculation [1]

(c) 1 mark for identifying the problem, 2 marks for the solution

Problem – doesn't deal correctly with remainder 10/a check digit of X

Solution – check Z for X as a final digit

– have a special case where check = 10

– accept where Check = 10 and F = X [3]

5

Riders	Reject	Height	Output
0	0		
1		1.4	
2		1.3	
	1	1.1	
3		1.3	
	2	1.0	
4		1.5	
5		1.2	
6		1.3	
7		1.4	
8		1.3	
			Ready to go 2

(1 mark) (1 mark) (1 mark) (1 mark)

[4]

7 (i) 1 mark for each improvement

use FOR … NEXT instead of REPEAT … UNTIL
Move PRINT to after the end of the loop
Add error checking to check that the value input is positive [3]

(ii) 3 marks maximum, 1 mark for each improvement correctly included.

Sample answer below

```
1 Total = 0
2       FOR Counter = 1 To 10
3         REPEAT
4           INPUT Num
5       UNTIL Num >0
6           Total = Total + Num
7  NEXT Counter
8 PRINT Total
```

[3]

6

Area	Tins	Height	Width	Doors	Windows
0	0	3	5	1	0
13.5		3	7	0	0
34.5		3	5	0	3
46.5		3	7	1	1
65		–1	0	0	0
	7				

(2 marks) (1 mark) (1 mark)
1 mark 0, 13.5
1 mark for rest

[4]

Question	Answer	Marks
11	1 mark for each error identified and suggested correction (the corrected code must be written in full) *Line* 2 *Correct code* `Counter = 0` (1) *Line* 7 *Correct code* `Total = Total + Number` // `Number + Total` (1) *Line* 8 *Correct code* `Counter = Counter + 1` // `1 + Counter` (1) *Line* 10 *Correct code* `Average = Total / Counter` // `Average = Total / 50` (1)	**4**

Question	Answer	Marks
8	Must match question. 2 marks for three suitable sub system names 1 mark for two suitable sub system names	**2**

Question	Answer	Marks
9	1 mark for each correct answer	**4**

Statements	Validation	Verification
To automatically check the accuracy of a bar code	✓	
To check if the data input is sensible	✓	
To check if the data input matches the data that has been supplied		✓
To automatically check that all required data fields have been completed	✓	

Question	Answer	Marks

Question	Answer	Marks
10		5

HighF	HighC	TempF	OUTPUT
–100	–100		
		68	
68	18	46	
68	18	50	
68	18	86	
86	27	65	
86	27	50	
86	27	40	
86	27	30	
86	27	–1	The highest temperature is, 86 Fahrenheit, 27 Celsius.

(1 Mark) (1 Mark) (1 Mark) (2 Marks – see below)

The literal correct output is “The highest temperature is, 86 Fahrenheit, 27 Celsius.”
1 mark for values 86 and 27, 1 mark for correct output words, spacing and punctuation.

Question	Answer	Marks
12		5

Weight	Reject	Total Weight	OUTPUT
	0	0	
13		13	
17		30	
26	1		
25		55	
5		60	
10		70	
15		85	
35	2		
20		105	
		85	Weight of items 85 Number of items rejected 2
(1mark)	(1 mark)	(1 mark to 1st 85) (1 mark 105, 85)	(1 mark)

Question	Answer	Marks

Question	Answer	Marks
13	**One** mark for each (max **three**) *10.00* boundary/erroneous data // the price should be rejected // value is out of range *9.99* boundary/extreme/normal data // the prices should be accepted // value is within normal range *ten* erroneous/abnormal data // input should be rejected // value is wrong type	**3**

Question	Answer	Marks
14		**5**

Digit(1)	Digit(2)	Digit(3)	Digit(4)	Digit(5)	Digit(6)	Digit(7)	Digit(8)	Sum	OUTPUT
5	7	0	1	2	3	4	6	44	GTIN-8
									57012346

Digit(1)	Digit(2)	Digit(3)	Digit(4)	Digit(5)	Digit(6)	Digit(7)	Digit(8)	Sum	OUTPUT
4	3	1	0	2	3	1	0	30	GTIN-8
									43102310

One mark for data entry – both sets of digits 1–7
One mark for both Digit(8)
One mark for each Sum (max **Two**)
One mark for both OUTPUT

Question	Answer	Marks
15	**One** mark per value and reason, max **3** Example 1.00 – boundary rejected//rejected (underweight) // out of range(1) 1.02 – normal // valid // accepted weight in range (1) 1.10 – abnormal // erroneous // invalid // rejected (overweight) (1)	**3**

Question	Answer	Marks
17(a)		3

Number1	Number2	Sign	Answer	OUTPUT
5	7	+	12	12
6	2	-	4	4
4	3	*	12	12
7	8	?	0	
0	0	/	(0)	

1 mark 1 mark 1 mark

Question	Answer	Marks
16(a)	Max **4** in total Any **3** from: • To ensure no changes are made on input / accuracy of transcription • Because the details do not have fixed, values or lengths to validate • Because there is no clear set of rules that can be used for validation Any **3** from: • The programmer could ask the contributor to type in each detail twice o and then check that both values are equal o If they are not equal then the input should be rejected • The programmer could ask the contributor to check the details on the screen o and confirm that they are correct / same as the original o or change them	4
16(b)	**One** mark for email and **one** mark for password Email – check for @ / format check / no spaces /valid characters // presence check // length check (not more than 254 characters) // uniqueness check Password – length check / numbers and letters etc. // uniqueness check not been used before // presence check	2

Question	Answer	Marks
	Section B	
18(a)	**One** mark for description **one** mark for example e.g. To test if the data entered is possible / reasonable A range check tests that data entered fits within specified values. Allow any correct validation check as an example	**2**
18(b)	**One** mark for description **one** mark for example e.g. To test if the data input is the same as the data that was intended to be input A double entry check expects each item of data to be entered twice and compares both entries to check they are the same. Allow any correct verification check as an example	**2**

Question	Answer	Marks
19(a)	Any **two** from: • Expects a number to be input • Checks if the number is greater than 100 • Outputs the result of the test • Specific output example	**2**

Question	Answer	Marks
20(a)		3

Max	Counter	Num	OUTPUT
–1000.00	0	6.30	
6.30	1	18.62	
18.62	2	50.01	
50.01	3	3.13	
50.01	4	2.05	
50.01	5	50.10	
50.10	6	40.35	
50.10	7	30.69	
50.10	8	0.85	
50.10	9	17.30	
50.10	10		50.10

1 mark 1 mark 1 mark

Question	Answer	Marks
21(a)	Range check	**1**
21(b)	**Two** from: • The entered number (`Value`) is being checked to see that it is not < 0 or not > 100 • If it is, it is rejected and the user has to enter another number / an error message is displayed • Otherwise the number is accepted, the word 'Accepted' is output along with the `Value`	**2**

Question	Answer	Marks
21(c)	**Value** | **OUTPUT** `Input a value between 0 and 100 inclusive` 200 — `Invalid value, try again` 300 — `Invalid value, try again` –1 — `Invalid value, try again` 50 — `Accepted: 50` 1 mark – `Value` column 1 mark – OUTPUT column first line 1 mark – OUTPUT column lines two to five	**3**

Question	Answer	Marks
22	**Validation** **Two** from: • automated checking • checking that data is reasonable / of a certain type • checking that data meets certain criteria Example 1 mark • range check // length check // type check // check digit etc. **Verification** **Two** from: • checking that data has not changed • during input to a computer • during transfer between computers / devices Example 1 mark • double entry // checking against original // visual check // use of checksum etc.	**6**

Question	Answer	Marks
23		4

Total	Count	Distinction	Mark	OUTPUT
0	0	0	50	
50	1	0	70	
120	2	0	65	
185	3	0	30	
215	4	0	95	
310	5	1	50	
360	6	1	55	
415	7	1	85	
500	8	2	65	
565	9	2	35	
600	10		–1	Number of Distinctions 2
				Average Mark 60

1 mark for Total and Count columns both correct.
1 mark for each correct column apart from Total and Count.
If no marks awarded allow 1 mark for initialisation of Total, Count and Distinction, set to zero.

Question	Answer	Marks
24(a)	1 mark for each error identified + suggested correction • `Low ← Count` should be `Low ← Number` • `Number > Low` should be `Number < Low` • `UNTIL Count = 99` should be `UNTIL Count > 99` or `UNTIL Count = 100` or `UNTIL Count >= 100` // `Count ← 1` should be `Count ← 0` • `PRINT "Largest Number is ", Number` should be `PRINT "Largest Number is ", High`	**4**

Question	Answer	Marks
25		4

Statements	Validation	Verification
Range check	✓	
Double entry		✓
Check digit	✓	
Presence check	✓	

1 mark for each correct row

Question	Answer	Marks
26		**4**

TreadReject	Count	Depth	OUTPUT
0	1		
	2	1.7	
	3	1.9	
1	4	1.4	
	5	1.8	
	6	2.0	
			Car is potentially roadworthy

TreadReject	Count	Depth	OUTPUT
0	1		
1	2	1.2	
	3	1.9	
2	4	1.4	
	5	1.8	
	6	2.4	
			Car is not roadworthy

1 mark for each correct pairs of columns.

Question	Answer	Marks
	Section B	

27(a)	1 mark for each error identified + suggested correction
	`Count ← 1` should be `Count ← 0` or `Count >= 500` should be `Count > 500`
	`AND` should be `OR`
	`Reject ← Reject - 1` should be `Reject ← Reject + 1`
	`Reject ← Reject/100` should be `Reject ← Reject/5` or `Reject * 100 / 500`
		4

Question	Answer	Marks
28		2

Statement	True (✓)	False (✓)
A structure diagram is a piece of code that is available throughout the structure of a program.		✓
A structure diagram shows the hierarchy of a system.	✓	
A structure diagram is another name for an array.		✓
A structure diagram shows the relationship between different components of a system.	✓	

Two marks for four correct rows.
One mark for three correct rows.

Question	Answer	Marks
29	**One** mark for each correct validation check (max **two**) • Range • Length • Type • Check Digit **One** mark for each correct related purpose (max **two**) e.g. • To make sure the data entered falls within a specific set of values • To make sure the data entered is no longer than specified • To make sure the data entered follows rules related to whether it is numbers of letters • To make sure an identification code entered is genuine or possible	**4**
29(b)	**One** mark for correct verification check (max **one**) • Double (data) entry • Visual check	**1**
29(c)	Any **two** correct statements (max **two**) e.g. • Validation checks if the data entered is possible/it cannot check if data has been entered correctly. • Verification checks if the data entered matches the data submitted for entry/ it does not check if data matches set criteria.	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>30</td><td>
<table>
<tr><th>Value</th><th>Calc1</th><th>Calc2</th><th>OUTPUT</th></tr>
<tr><td>50</td><td>25</td><td>16</td><td></td></tr>
<tr><td>33</td><td>16</td><td>11</td><td></td></tr>
<tr><td>18</td><td>9</td><td>6</td><td>18</td></tr>
<tr><td>15</td><td>7</td><td>5</td><td></td></tr>
<tr><td>30</td><td>15</td><td>10</td><td>30</td></tr>
<tr><td>-1</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
One mark for each correct column (max four)
</td><td>4</td></tr>
<tr><td>30(b)</td><td>Any two correct statements e.g.
 The program outputs a value
 That is divisible by 6 // 2 and 3</td><td>2</td></tr>
</table>

Question	Answer	Marks
31	**One** mark for correct input (`PointsWon` and `PointsLost`) **One** mark for correct calculations (`Difference`) **One** mark for correct output	**3**

PointsWon	PointsLost	Difference	OUTPUT
5000	4474	526	Keep on trying
6055	2000	4055	Well done move up
7900	9800	-1900	Sorry move down
3000	2150	850	Keep on trying
-1	6700		

Question	Answer	Marks
32(a)	**One** mark for error and correction Line 1 `HighestMark ← 0` Line 7 `INPUT Mark[Count]` Line 10 `HighestMarkStudents ← HighestMarkStudents + 1` Line 14 `HighestMark ← Mark[Count]`	**4**

Question	Answer	Marks
35	Line 2 and Line 4 errors - **One** mark for each correct identification **and** correction of error Error 1 line number: Line 2 Correction: `REPEAT` Error 2 line number: Line 4 Correction: `IF Number < 0 OR Number > 499` Line 8 error - **One** mark for correct identification of error and **one** mark for **each** correction of error Error 2 line number: Line 8 Correction: `UNTIL Number >= 0 AND Number <= 499`	**6**

Question	Answer	Marks
33(a)	**One** mark for each correct check (max **two**) - Length (check) - Type Check - Format Check	**2**
33(b)	**One** mark for each suitable piece of test data and **one** mark for each relevant reason (max **four**) - LL9999LL999 - Too long - 5678987 - All numeric - CB12EU - No space is present	**4**

Question	Answer	Marks
34		**5**

Flag	Number	Divisor	Value	OUTPUT
False	5	2	2	
		3		
				5 is prime
False	6	2	3	
True		3	2	
True		4		
False	8	2	4	
True		3	2	
True		4	2	
		5		
False	0			

One mark for each correct column