

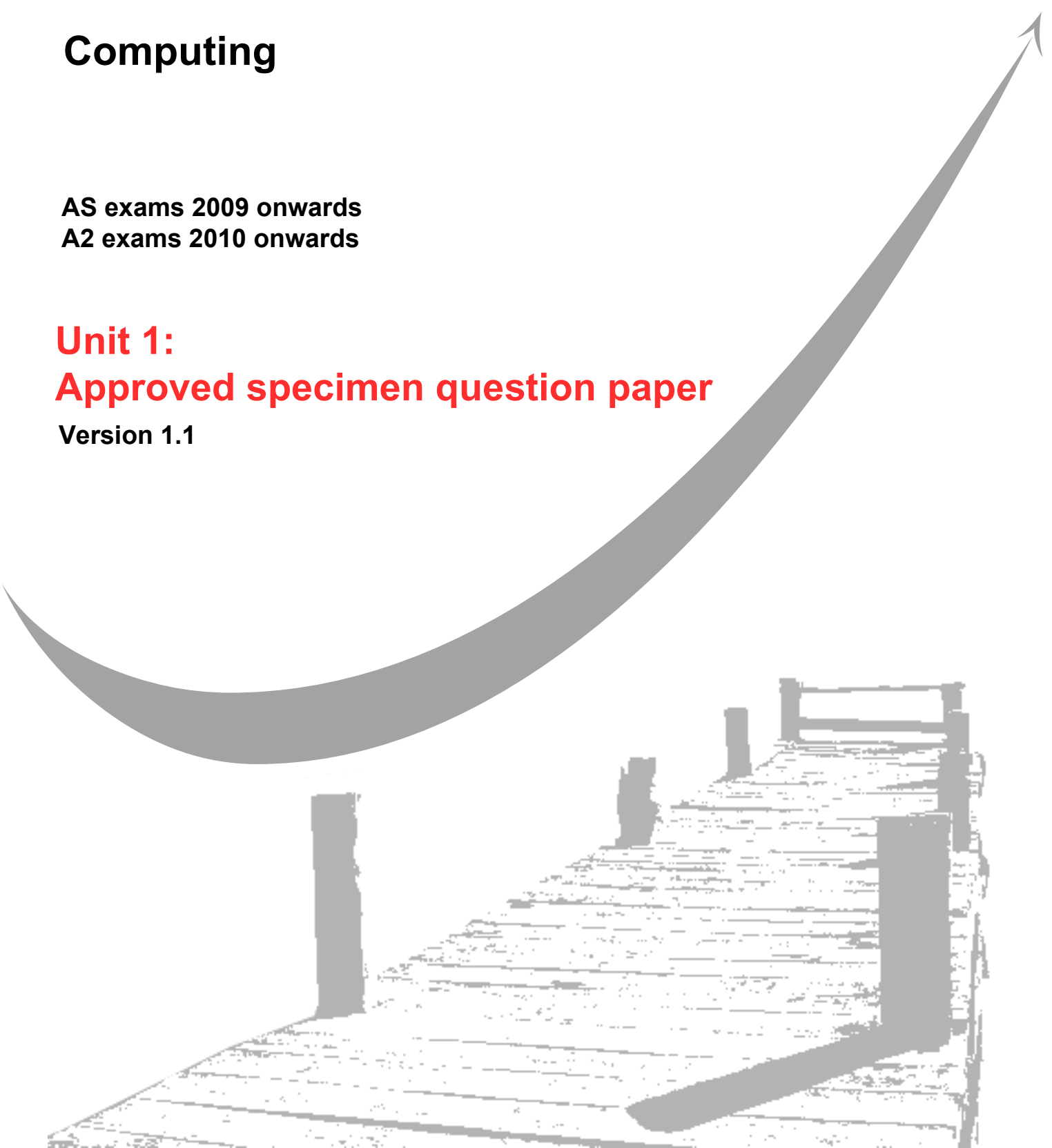
GCE
AS and A Level

Computing

AS exams 2009 onwards
A2 exams 2010 onwards

Unit 1:
Approved specimen question paper

Version 1.1



General Certificate of Education
Advanced Subsidiary
Specimen Paper for 2009 and 2010



COMPUTING

COMP1 Problem Solving, Programming, Data Representation and Practical Exercise

Date Time

You will need to refer to the pre-release material.
No calculator allowed.

Time allowed: 2 hours

Instructions

- Answer **all** questions.
- You will need access to:
 - a computer
 - a printer
 - appropriate software
 - electronic versions of the pre-release material (skeleton program and test data).
- At the end of the examination make sure your **centre number** and **candidate name and number** is shown clearly on every print-out.
- Cross through any work you do not want marked.

Information

- The maximum mark for this paper is 100.
- The marks for questions are shown in brackets.
- No extra time is allowed for printing and collating.

At the end of the examination

- Hand in **both** your completed Cover Sheet **and** all your print-outs to the invigilator.

Warning

- It may not be possible to issue a result for this unit if your details are not printed on every page.

Note: Specimen material for Unit 1 using alternative programming languages will be available shortly.

Answer **all** questions.

- 1** Binary patterns can be interpreted in a number of different ways. A main memory location contains the binary pattern 10011100. What is its denary value if it represents:
- (a) An unsigned integer? *(1 mark)*
 - (b) A two's complement integer? *(1 mark)*
 - (c) An unsigned fixed point number with the binary point after the 4th bit? *(2 marks)*
- 2**
- (a) (i) How can a black and white image be represented as a bit-mapped graphic? *(2 marks)*
 - (ii) What change needs to be made to (a)(i) to represent a 256-colour image? *(1 mark)*
 - (b) When sound from a microphone is recorded, how is it converted into a form which can be stored in a computer system? *(2 marks)*
- 3** Why is the Hexadecimal number system used in a computer context? *(1 mark)*

The following questions refer to the program code you have been given.

- 4 (a) (i) Give the line number where a global variable is defined. *(1 mark)*
(ii) Give the line number where a local variable is defined. *(1 mark)*
(b) (i) Give the line number where a function is declared. *(1 mark)*
(ii) Explain how a function differs from a procedure. *(1 mark)*
- 5 (a) Give the name of **one** 'call by reference' parameter. *(1 mark)*
(b) Give the name of **one** 'call by value' parameter. *(1 mark)*
(c) Explain how a 'call by reference' parameter differs from a 'call by value' parameter. *(3 marks)*
- 6 This question refers to the function Binary.
- (a) Explain what line 57 does. *(1 mark)*
(b) What role does the variable HexDigit perform? *(1 mark)*
(c) Explain what line 79 does. *(3 marks)*
(d) Complete the Case statement and copy the code of the whole Case statement into your answer. *(4 marks)*
- 7 This question refers to the main program body.
The program should show the menu repeatedly to the user until the user enters choice 4. Then the program should end.
- (a) Write the necessary code to show the menu repeatedly. *(1 mark)*
(b) Write the necessary code so that the program ends when the user enters choice 4.
Copy the code of the whole program body into your answer *(2 marks)*
- 8 This question refers to the procedure GetResponse.
The program should not allow the user to proceed until a valid option has been entered.
Add the code required to validate the options the user enters.
Copy the code of the whole procedure GetResponse into your answer. *(3 marks)*

-
- 9 This question refers to the function Binary.
Currently the program relies on the input string to be valid hexadecimal numbers.
- (a) Study the algorithm below.
Result <-- EmptyString
ErrorFound <-- False
For each HexDigit in InputString
If HexDigit is a valid hexadecimal digit
Then Get binary equivalent and append to Result
Else set ErrorFound to True
If ErrorFound
Then return an error message instead of binary code.
Adapt the code for the function Binary to reflect the solution given in the above algorithm.
- (b) Copy the whole code for the function Binary into your answer. *(5 marks)*
- 10 This question refers to testing the function Binary.
To test a routine adequately, normal, boundary and erroneous data need to be used.
Look at your version of the function Binary.
- (a) (i) Give **three** suitable values of normal test data. *(5 marks)*
(ii) Take a copy of the output screen when you run your program using these values. *(1 mark)*
- (b) (i) Give **two** values of erroneous test data. *(4 marks)*
(ii) Take a copy of the output screen when you run your program using these values. *(1 mark)*
- (c) An example of boundary data might be the empty string.
This should not cause an error but give the empty string as a result.
Write the instructions necessary to handle this boundary value.
Copy the entire code for your Binary function into your answer. *(4 marks)*
- 11 This question refers to the procedure ConvertFile.
Currently the procedure converts hexadecimal numbers read from the text file HexData.dat and displays the binary equivalents on the console.
The procedure should save the binary equivalents in a textfile called BinaryData.dat.
- (a) Write the code to do this.
Copy the code of the whole procedure ConvertFile into your answer. *(7 marks)*
- (b) Run your program with the datafile supplied.
Attach the resulting textfile to your answer. *(4 marks)*

12 This question refers to the procedure DisplayFile.

(a) Write the code needed so the user can type in the file name of the file to be displayed. *(2 marks)*

(b) The binary file is to be interpreted as an image.
Each 0 is to be a space and each 1 is to be a @.

(i) Write an algorithm to produce a solution to the above. *(5 marks)*

(ii) Edit the procedure DisplayFile to reflect your solution in (i) above.
Copy the code of your whole procedure DisplayFile into your answer. *(5 marks)*

(iii) Run program option Display File with the file BinaryData.dat produced in question 11. Take a copy of the screen showing your output and paste it into your answer. *(4 marks)*

13 This question expands the functionality of the program.

(a) Outline the changes needed to give the user the option to convert a hex number into denary. *(8 marks)*

(b) Implement the changes outlined above. And copy your whole code into your answer. *(7 marks)*

14 Save your entire program and attach it to your answer. *(4 marks)*

END OF QUESTIONS

PRE-RELEASE MATERIAL – SKELETON PROGRAM

Line Number	Program statements
1	Program Example;
2	
3	{ \$APPTYPE CONSOLE }
4	
5	// written by Principal Examiner
6	
7	// completed by
8	// Centre Number
9	// Candidate Number
10	
11	// some statements are incomplete and
12	// therefore do not do what they should do
13	
14	// This program is designed to
15	// accept a user's menu choice to
16	// 1 - read in a single hexadecimal number
17	// and display the binary equivalent
18	// 2 - read in a text file of hexadecimal numbers and
19	// produce a text file of their binary equivalent
20	// 3 - display the content of a text file
21	// 4 - finish running the program
22	
23	
24	Uses
25	SysUtils;
26	
27	Var Choice: Integer;
28	Finished: Boolean;
29	
30	Procedure ShowMenu;
31	Begin
32	Writeln (' Please choose an option');
33	Writeln;
34	Writeln (' 1 - Convert a Hex number to Binary');
35	Writeln (' 2 - Convert a text file of Hex numbers');
36	Writeln (' 3 - Display text file');
37	Writeln (' 4 - exit program');
38	Writeln;
39	End; // of Procedure ShowMenu
40	
41	Procedure GetResponse(Var Response: Integer);
42	Begin
43	Write (' Enter option number: ');
44	Readln(Response);
45	End; // of Procedure GetResponse
46	

```
47     Function Binary(Hex: String):String;  
48     Var HexDigit, NoOfHexDigits : Integer;  
49         ThisHexDigit: Char;  
50         BinaryEquivalent : String[4];  
51     Begin  
52         Result := '';  
53         NoOfHexDigits := Length(Hex);  
54         For HexDigit := 1 To NoOfHexDigits  
55             Do  
56                 Begin  
57                     ThisHexDigit := Hex[HexDigit];  
58                     If ThisHexDigit In ['0'..'9','A'..'F']  
59                         Then  
60                             Case ThisHexDigit Of  
61                                 '0': BinaryEquivalent := '';  
62                                 '1': BinaryEquivalent := '';  
63                                 '2': BinaryEquivalent := '';  
64                                 '3': BinaryEquivalent := '';  
65                                 '4': BinaryEquivalent := '';  
66                                 '5': BinaryEquivalent := '';  
67                                 '6': BinaryEquivalent := '';  
68                                 '7': BinaryEquivalent := '';  
69                                 '8': BinaryEquivalent := '';  
70                                 '9': BinaryEquivalent := '';  
71                                 'A': BinaryEquivalent := '';  
72                                 'B': BinaryEquivalent := '';  
73                                 'C': BinaryEquivalent := '';  
74                                 'D': BinaryEquivalent := '';  
75                                 'E': BinaryEquivalent := '';  
76                                 'F': BinaryEquivalent := '';  
77                             End // of Case  
78                             Else ;  
79                             Result:= Result + BinaryEquivalent;  
80                         End; // of For loop  
81                 End; // of Function Binary  
82  
83     Procedure ConvertNumber;  
84     Var Hexadecimal, Converted: String;  
85     Begin  
86         Write ('Enter a Hexadecimal number: ');  
87         Readln (Hexadecimal);  
88         Converted := Binary(Hexadecimal);  
89         Writeln (Converted);  
90     End; // of Procedure ConvertNumber  
91  
92     Procedure ConvertFile;  
93     Var FileNameIn : String;  
94         HexFile : TextFile;  
95         HexNumber, BinaryNumber : String;  
96     Begin  
97         FileNameIn := 'HexData.dat';
```

```
98     AssignFile (HexFile, FileNameIn);
99     Reset (HexFile);
100    While Not EOF (HexFile)
101        Do
102            Begin
103                Readln(HexFile, HexNumber);
104                BinaryNumber := Binary(HexNumber);
105                Writeln(BinaryNumber);
106            End;
107        CloseFile (HexFile);
108    End; // of Procedure ConvertFile
109
110    Procedure DisplayFile;
111    Var FileName: String; Bit: Char;
112        BinaryFile: TextFile;
113    Begin
114        FileName := 'BinaryData.dat';
115        AssignFile (BinaryFile, FileName);
116        Reset (BinaryFile);
117        While Not EOF (BinaryFile)
118            Do
119                Begin
120                    Read(BinaryFile, Bit);
121                    Write(Bit);
122                End;
123            CloseFile (BinaryFile);
124        End; // of Procedure DisplayFile
125
126    //***** Main Program Body *****
127    Begin
128        Finished := False;
129
130        ShowMenu;
131        GetResponse(Choice);
132        Case Choice Of
133            1: ConvertNumber;
134            2: ConvertFile;
135            3: DisplayFile;
136            4: // end program
137        End; //of Case
138
139    End.
```

PRE-RELEASE MATERIAL – TEST DATA

00000000000
000001110000
000010001000
000010001000
000001110000
001000100010
000100100100
000010101000
000001110000
000000100000
000000100000
000000100000
000001010000
000010001000
000100000100
00000000000