Version 1.0

General Certificate of Education
Advanced Subsidiary
Specimen Pre-release material for 2009 and 2010
Month Year

COMPUTING

COMP1/PR Problem Solving, Programming, Data Representation and Practical Exercise

Date Time

Pre-release material – Instructions for Candidates: Program Language PHP

To be given to candidates on or after 1 April XXXX

Information

- There are 2 parts to this pre-release material:
  - Section A: Skeleton Program
  - Section B: Test Data.
- This material will be re-printed in the examination paper.
- You are advised to familiarise yourself with this material before the examination.
- You will use both the program and test data in the examination and your teacher will provide you with access to these electronically at the start of the examination.
- You can view/download this material on the AQA Website: www.aqa.org.uk/xxxxxx

SECTION A – Skeleton Program

```
 Line     Program statements
Number
   1      <?php
   2
   3      // some statements are incomplete and
   4      // therefore do not do what they should do
   5
   6      // This program is designed to
   7      // accept a user's menu choice to
   8      // 1 - read in a single hexadecimal number
   9      //     and display the binary equivalent
  10      // 2 - read in a text file of hexadecimal numbers and
  11      //     produce a text file of their binary equivalent
  12      // 3 - display the content of a text file
  13      // 4 - finish running the program
  14
  15      function ShowMenu()
  16      {
  17        fwrite(STDOUT, "Please choose an option: \n\n");
  18        fwrite(STDOUT, "1 : Convert a Hex number to Binary\n");
  19        fwrite(STDOUT, "2 : Convert a text file of Hex numbers\n");
  20        fwrite(STDOUT, "3 : Display text file\n");
  21        fwrite(STDOUT, "4 : Exit program\n\n");
  22      }   // end of function ShowMenu
  23
  24      function GetResponse()
  25      {
  26        fwrite(STDOUT, "Enter option number: ");
  27        $Response = intval(trim(fgets(STDIN)));
  28        return $Response;
  29      }  // end of function GetResponse
  30
  31      function Binary($Hex)
  32      {
  33        $Result = "";
  34        $BinaryEquivalent = "";
  35        $NoOfHexDigits = strlen($Hex);
  36        for ($HexDigit = 0; $HexDigit < $NoOfHexDigits; $HexDigit++)
  37        {
  38          $ThisHexDigit = substr($Hex, $HexDigit, 1);
  39          if (($ThisHexDigit >='0'&& $ThisHexDigit <='9') ||
  40                 ($ThisHexDigit >='A' && $ThisHexDigit <='F'))
  41          {
  42            switch ($ThisHexDigit)
  43            {
  44              case '0' : $BinaryEquivalent = ""; break;
  45              case '1' : $BinaryEquivalent = ""; break;
  46              case '2' : $BinaryEquivalent = ""; break;
  47              case '3' : $BinaryEquivalent = ""; break;
  48              case '4' : $BinaryEquivalent = ""; break;
  49              case '5' : $BinaryEquivalent = ""; break;
  50              case '6' : $BinaryEquivalent = ""; break;
  51              case '7' : $BinaryEquivalent = ""; break;
  52              case '8' : $BinaryEquivalent = ""; break;
  53              case '9' : $BinaryEquivalent = ""; break;
  54              case 'A' : $BinaryEquivalent = ""; break;
  55              case 'B' : $BinaryEquivalent = ""; break;
  56              case 'C' : $BinaryEquivalent = ""; break;
```

```
57              case 'D' : $BinaryEquivalent = ""; break;
58              case 'E' : $BinaryEquivalent = ""; break;
59              case 'F' : $BinaryEquivalent = ""; break;
60            }
61          }
62          else ;
63          $Result = $Result.$BinaryEquivalent;
64        }
65        return $Result;
66      }  // end of function Binary
67
68      function ConvertNumber()
69      {
70        fwrite(STDOUT, "Enter a Hexadecimal number: ");
71        $HexNumber = trim(fgets(STDIN));
72        $Converted = Binary($HexNumber);
73        fwrite(STDOUT, $HexNumber." = ".$Converted."\n");
74      }   // end of function ConvertNumber
75
76      function ConvertFile()
77      {
78        $FileNameIn = "HexData.dat";
79        $HexFile = fopen($FileNameIn, "r");
80        while (! feof($HexFile))
81        {
82          $HexNumber = fgets($HexFile, 2);
83          $BinaryNumber = Binary($HexNumber);
84          fwrite(STDOUT, $BinaryNumber. "\n");
85        }
86        fclose($HexFile);
87      }  // end of function ConvertFile
88
89      function DisplayFile()
90      {
91        $FileName = 'BinaryData.dat';
92        $BinaryFile = fopen($FileName, "r");
93        while (! feof($BinaryFile))
94        {
95          $Bit = fgetc($BinaryFile);
96          fwrite(STDOUT, $Bit);
97        }
98        fclose($BinaryFile);
99        fwrite(STDOUT, "\n\n");
100     }  // end of function DisplayFile
101
102     ShowMenu();
103     $Choice = GetResponse();
104     switch ($Choice)
105     {
106           case 1 : ConvertNumber(); break;
107           case 2 : ConvertFile(); break;
108           case 3 : DisplayFile(); break;
109           case 4 : break;   // end program
110     }   // end switch
111
112     ?>
```

SECTION B – Test Data

000000000000
000001110000
000010000100
000010001000
000001110000
001000100010
000100100100
000010101000
000001110000
000000100000
000000100000
000000100000
000001010000
000010001000
000100000100

END OF PRE-RELEASE MATERIAL