Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/42**

Paper 42 Practical **May/June 2022**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the May/June 2022 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

This document consists of **36** printed pages.

© UCLES 2022

PUBLISHED

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

© UCLES 2022

GENERIC MARKING PRINCIPLE 5: Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).
GENERIC MARKING PRINCIPLE 6: Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

© UCLES 2022

Question	Answer	Marks

1(a)	1 mark per mark point
	• declaring array `StackData` and pointer `StackPointer` as (global data structures)
	• `StackData` has 10 integer elements
	• `StackPointer` initialised to 0
	
	Example program code:
	VB.NET
	`Dim StackData(9) As Integer`
	`Dim StackPointer As Integer`
	`Sub Main()`
	`  StackPointer = 0`
	`end Sub`
	
	Python
	`global StackData #integer`
	`global StackPointer`
	`StackData = [0,0,0,0,0,0,0,0,0,0] #integer`
	`StackPointer = 0`
	
	Java
	`import java.util.Scanner;`
	`class Question1{`
	`  public static Integer[] StackData;`
	`  public static Integer StackPointer;`
	`  public static void main(String args[]){`
	`  StackData = new Integer[10];`
	`  StackPointer = 0;`
	`  }`
	`}`
		3

Question	Answer	Marks

1(b)	1 mark per mark point
	• procedure header with sensible identifier (and end where appropriate)
	• outputting `StackPointer`
	• outputting all **10** elements in array
	
	Example program code:
	VB.NET
	`Sub PrintArray()`
	`  Console.WriteLine(StackPointer)`
	`  For x = 0 To 9`
	`  Console.WriteLine(StackData(x))`
	`  Next`
	`End Sub`
	
	Python
	`def PrintArray():`
	`  global StackData`
	`  global StackPointer`
	`  print(StackPointer)`
	`  for x in range (0, 10):`
	`  print(StackData[x])`
	
	Java
	`public static void PrintArray(){`
	`  System.out.println(StackPointer);`
	`  for(int x = 0; x < 10 ;x++){`
	`  System.out.println(StackData[x]);`
	`  }`
	`}`
		3

Question	Answer	Marks

1(c)	1 mark per mark point
	• function `Push()` taking an integer parameter
	
	• checking if stack is full …
	• …and returning `FALSE`
	
	• (if not full) **storing parameter** to stack at `StackPointer` …
	• …incrementing `StackPointer`
	• …returning `TRUE`
	
	Example program code:
	VB.Net
	`Function Push(DataToPush)`
	`  If StackPointer = 10 Then`
	`  Return False`
	`  Else`
	`  StackData(StackPointer) = DataToPush`
	`  StackPointer = StackPointer + 1`
	`  Return True`
	`  End If`
	`End Function`
	
	Python
	`def Push(DataToPush):`
	`  global StackData`
	`  global StackPointer`
	`  if StackPointer == 10:`
	`  return False`
	`  else:`
	`  StackData[StackPointer] = DataToPush`
	`  StackPointer = StackPointer + 1`
	`  return True`
		6

Question	Answer	Marks

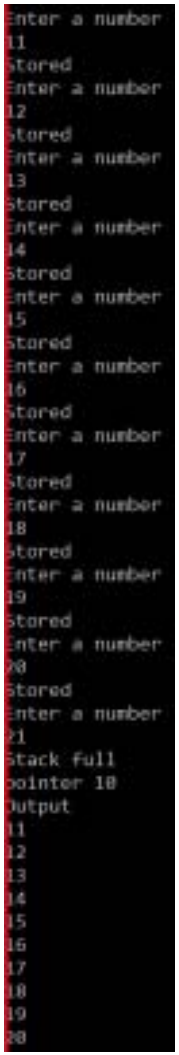
1(c)	**Java**
	`public static Boolean Push(Integer DataToPush){`
	`  if(StackPointer == 10){`
	`  return false;`
	`  }else{`
	`  StackData[StackPointer] = DataToPush;`
	`  StackPointer = StackPointer + 1;`
	`  return true;`
	`  }`
	`}`	

Question	Answer	Marks

1(d)(i)	1 mark per mark point
	• Inputting 11 numbers …
	• …calling `Push()` with each number input as a parameter …
	• …outputting appropriate message if `TRUE` returned
	• …outputting appropriate message if `FALSE` returned
	
	• Calling their output procedure **after** all **11** additions
	
	Example program code:
	VB.NET
	`Sub Main()`
	`  StackPointer = 0`
	`  Dim TempNumber As Integer`
	`  For x = 0 To 10`
	`  Console.WriteLine("Enter a number")`
	`  TempNumber = Console.ReadLine()`
	
	`  If Push(TempNumber) Then`
	`  Console.WriteLine("Stored")`
	`  Else`
	`  Console.WriteLine("Stack full")`
	`  End If`
	`  Next`
	`  PrintArray()`
	`  Console.ReadLine()`
	`End Sub`
		5

Question	Answer	Marks

1(d)(i)	**Python**
	`#main`
	`StackPointer = 0`
	`StackData = [0,0,0,0,0,0,0,0,0,0]`
	`for x in range(0, 11):`
	`    TempNumber = int(input("Enter a number"))`
	
	`    if Push(TempNumber) == True:`
	`        print("Stored")`
	`    else:`
	`        print("Stack full")`
	`PrintArray()`
	
	Java
	`public static void main(String[] args){`
	`   StackData = new Integer[10];`
	`   StackPointer = 0;`
	`   Integer TempNumber = 0;`
	`   for(int x = 0; x < 10; x++){`
	`      System.out.println("Enter a number");`
	`      Scanner scanner = new Scanner(System.in);`
	`      TempNumber = Integer.parseInt(scanner.nextLine());`
	`      if(Push(TempNumber)){`
	`          System.out.println("Stored");`
	`      }else{`
	`          System.out.println("Stack full");`
	`           }`
	`      }`
	`    PrintArray();`
	`}`	

Question	Answer	Marks
1(d)(ii)	1 mark for inputting all 11 numbers, message for first 10 saying added (11 to 20), message stating 11th number stating stack full. Full array contents output (11 12 13 14 15 16 17 18 19 20). e.g. 	**1**

Question	Answer	Marks

1(e)(i)	1 mark per mark point
	• `Pop()` function header (and close where appropriate) **and** returning a number in all possible situations
	• checking if stack is empty (`StackPointer` is `0`) **and** returning `-1`
	• (otherwise) **accessing** the item at the top of the stack …
	• …decrementing the stack pointer
	• …returning the item removed
	
	Example Program code:
	VB.NET
	`Function Pop()`
	`  Dim ReturnData As Integer`
	`  If StackPointer = 0 Then`
	`  Return -1`
	`  Else`
	`  ReturnData = StackData(StackPointer – 1)`
	`  StackPointer = StackPointer - 1`
	`  Return ReturnData`
	`  End If`
	`End Function`
	
	Python
	`def Pop():`
	`  global StackData`
	`  global StackPointer`
	`  if StackPointer == 0:`
	`  return -1`
	`  else:`
	`  ReturnData = StackData[StackPointer - 1]`
	`  StackPointer = StackPointer - 1`
	`  return ReturnData`
		5

Question	Answer	Marks

1(e)(i)	**Java**
	`public static Integer Pop(){`
	`  Integer ReturnData = 0;`
	`  if(StackPointer == 0){`
	`  return -1;`
	`  }else{`
	`  ReturnData = StackData[StackPointer - 1];`
	`  StackPointer = StackPointer - 1;`
	`  return ReturnData;`
	`  }`
	`}`	

1(e)(ii)	1 mark per mark point
	• output of before removed with 11 inputs
	• output of stack (after, this could be 11–20, 11–18, or 11–18 then 'null' 'null's)
	e.g.
	`Enter a number11`
	`Stored`
	`Enter a number12`
	`Stored`
	`Enter a number13`
	`Stored`
	`Enter a number14`
	`Stored`
	`Enter a number15`
	`Stored`
	`Enter a number16`
	`Stored`
	`Enter a number17`
	`Stored`
	`Enter a number18`
	`Stored`
	`Enter a number19`
	`Stored`
	`Enter a number20`
	`Stored`
	`Enter a number21`
	`Stack full`
	`8`
	`Output`
	`11`
	`12`
	`13`
	`14`
	`15`
	`16`
	`17`
	`18`
	`19`
	`20`
	`11`
	`12`
	`13`
	`14`
	`15`
	`16`
	`17`
	`18`
	`19`
	`20`
		2

Question	Answer	Marks

2(a)	1 mark per mark point
	• in main local 2D array declared …
	• … with 10 × 10 integer elements
	• initialising **all** array elements to a number…
	• …that is random between 1 and 100 (allow inclusive or exclusive)
	
	Example program code:
	VB.NET
	`Sub Main()`
	`  Dim Random As New Random`
	`  Dim ArrayData(10, 10) As Integer`
	`  For x = 0 To 9`
	`  For y = 0 To 9`
	`  ArrayData(x, y) = Random.Next(1, 100)`
	`  Next`
	`  Next`
	`  Console.ReadLine()`
	`End Sub`
	
	Python
	`import random`
	`#main`
	`ArrayData= [[0]*10 for i in range(10)] #integer`
	`for x in range(0, 10):`
	`  for y in range(0,10):`
	`  ArrayData[x][y] = random.randint(1, 100)`
		4

Question	Answer	Marks

2(a)	**Java**
	`import java.util.Scanner;`
	`import java.util.Random;`
	
	`class Question2{`
	`  public static Integer[][] ArrayData;`
	`  public static void main(String args[]){`
	`  Random Rand = new Random();`
	`  ArrayData = new Integer[10][10];`
	`  for(int x=0; x < 10; x++){`
	`  for(int y = 0; y < 10; y++){`
	`  ArrayData[x][y] = Rand.nextInt(100);}`
	`  }`
	`  }`
	`}`	

Question	Answer	Marks

2(b)(i)	1 mark per mark point
	• 1st outer loop (dimension 1)
	• 2nd loop (dimension 2)
	• inner for loop for all second dimension
	
	• Selection statement …
	• …swapping the numbers correctly
	
	Example program code:
	VB.NET
	`Dim TempNumber As Integer`
	`Dim ArrayLength As Integer = 10`
	`For X = 0 To ArrayLength - 1`
	`  For Y = 0 To ArrayLength - 2`
	`    For Z = 0 To ArrayLength - Y - 2`
	`      if ArrayData(X, Z) > ArrayData(X, Z + 1) then`
	`        TempNumber = ArrayData(X, Z)`
	`        ArrayData(X, Z) = ArrayData(X, Z+1)`
	`        ArrayData(X, Z + 1) = TempNumber`
	`      end if`
	`    Next Z`
	`  Next Y`
	`Next X`
	
	Python
	`ArrayLength = 10`
	`for X in range(0, ArrayLength):`
	`    for Y in range(0, ArrayLength-1):`
	`        for Z in range(0, ArrayLength - Y - 1):`
	`            if(ArrayData[X][Z] > ArrayData[X][Z+1]):`
	`                TempNumber = ArrayData[X][Z]`
	`                ArrayData[X][Z] = ArrayData[X][Z+1]`
	`                ArrayData[X][Z+1] = TempNumber`
	Accept for MP5:
	`ArrayData[X][Z], ArrayData[X][Z+1] = ArrayData[X][Z+1], ArrayData[X][Z]`
		5

Question	Answer	Marks

2(b)(i)	**Java**
	`Integer ArrayLength = 10;`
	`for(int X = 0; X < ArrayLength; X++){`
	`  for(int Y = 0; Y < ArrayLength; Y++){`
	`  for(int Z = 0; Z < ArrayLength - Y - 1; Z++){`
	`  if(ArrayData[X][Z] > ArrayData[X][Z + 1]){`
	`  TempNumber = ArrayData[X][Z];`
	`  ArrayData[X][Z] = ArrayData[X][Z+1];`
	`  ArrayData[X][Z + 1] = TempNumber;`
	`  }`
	`  }`
	`  }`
	`}`	

Question	Answer	Marks
2(b)(ii)	1 mark per mark point • procedure header (and end where appropriate) • Outputting all 10 × 10 values with each 2nd dimension on a complete line • Calling procedure before and after bubble sort Example program code: **VB.NET**	**3**

```
Sub Main()
  Dim random As New Random
  Dim ArrayData(10, 10) As Integer
  For x = 0 To 9
    For y = 0 To 9
      ArrayData(x, y) = random.Next(1, 100)
    Next
  Next
  Console.WriteLine("before")
  printarray(ArrayData)

  Dim TempNumber As Integer
  Dim ArrayLength As Integer = 10
  For X = 0 To ArrayLength - 1
    For Y = 0 To ArrayLength - 2
      For Z = 0 To ArrayLength - Y - 2
        if ArrayData(X, Z) > ArrayData(X, Z + 1) then
          TempNumber = ArrayData(X, Z)
          ArrayData(X, Z) = ArrayData(X, Z+1)
          ArrayData(X, Z + 1) = TempNumber
        end if
      Next Z
    Next Y
  Next X

  Console.WriteLine("after")
  printarray(ArrayData)
  Console.ReadLine()
End Sub
```

Question	Answer	Marks

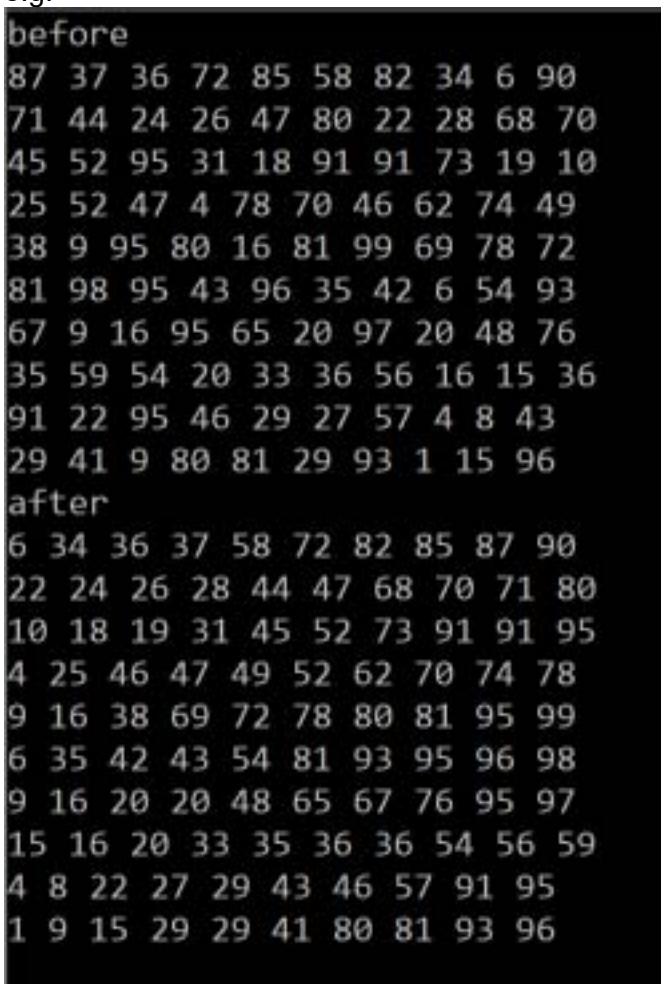
2(b)(ii)	**Python**
	`import random`
	
	`def Printarray(ArrayData):`
	`  for x in range(0, 10):`
	`  for y in range(0, 10):`
	`  print(ArrayData[x][y], " ", end='')`
	`  print("")`
	`#main`
	`ArrayData= [[0]*10 for i in range(10)] #integer`
	`for x in range(0, 10):`
	`  for y in range(0,10):`
	`  ArrayData[x][y] = random.randint(1, 100)`
	`print("Before")`
	`printarray(ArrayData)`
	`ArrayLength = 10`
	`for X in range(0, ArrayLength):`
	`  for Y in range(0, ArrayLength):`
	`  for Z in range(0, ArrayLength - Y - 1):`
	`  if(ArrayData[X][Z] > ArrayData[X][Z+1]):`
	`  TempNumber = ArrayData[X][Z]`
	`  ArrayData[X][Z] = ArrayData[X][Z+1]`
	`  ArrayData[X][Z+1] = TempNumber`
	
	`print("After")`
	`Printarray(ArrayData)`	

Question	Answer	Marks

2(b)(ii)	**Java**
	`import java.util.Scanner;`
	`import java.util.Random;`
	
	`class Question2{`
	
	`  public static Integer[][] ArrayData;`
	
	`  public static void printArray(Integer[][] theArrayData){`
	`    for(int x = 0; x < 10; x++){`
	`      for(int y = 0; y < 10; y++){`
	`        System.out.printf(theArrayData[x][y] + " ");`
	`        }`
	`      System.out.println();`
	`        }`
	`   }`
	
	`  public static void main(String args[]){`
	`  Random rand = new Random(){`
	`    ArrayData = new Integer[10][10];`
	`    for(int x=0; x < 10; x++){`
	`      for(int y = 0; y < 10; y++){`
	`        ArrayData[x][y] = rand.nextInt(100);`
	`      }`
	`    }`
	`    Integer TempNumber = 0;`
	
	`    System.out.println("Before");`
	`    printArray(ArrayData);`
	`    Integer ArrayLength = 10;`
	`      for(int X = 0; X < ArrayLength; X++){`
	`        for(int Y = 0; Y < ArrayLength; Y++){`
	`          for(int Z = 0; Z < ArrayLength - Y - 1; Z++){`
	`              if(ArrayData[X][Z] > ArrayData[X][Z + 1]){`	

Question	Answer	Marks

2(b)(ii)	`TempNumber = ArrayData[X][Z];`
	`ArrayData[X][Z] = ArrayData[X][Z+1];`
	`ArrayData[X][Z + 1] = TempNumber;`
	`}`
	`}`
	`}`
	`}`
	`System.out.println("After");`
	`PrintArray(ArrayData);`
	
	`}`
	`}`	

Question	Answer	Marks
2(b)(iii)	1 mark for output showing array unsorted and then sorted on 1 of the dimensions e.g.  before 87 37 36 72 85 58 82 34 6 90 71 44 24 26 47 80 22 28 68 70 45 52 95 31 18 91 91 73 19 10 25 52 47 4 78 70 46 62 74 49 38 9 95 80 16 81 99 69 78 72 81 98 95 43 96 35 42 6 54 93 67 9 16 95 65 20 97 20 48 76 35 59 54 20 33 36 56 16 15 36 91 22 95 46 29 27 57 4 8 43 29 41 9 80 81 29 93 1 15 96 after 6 34 36 37 58 72 82 85 87 90 22 24 26 28 44 47 68 70 71 80 10 18 19 31 45 52 73 91 91 95 4 25 46 47 49 52 62 70 74 78 9 16 38 69 72 78 80 81 95 99 6 35 42 43 54 81 93 95 96 98 9 16 20 20 48 65 67 76 95 97 15 16 20 33 35 36 36 54 56 59 4 8 22 27 29 43 46 57 91 95 1 9 15 29 29 41 80 81 93 96	**1**

© UCLES 2022

Question	Answer	Marks

2(c)(i)	1 mark for each completed statement (6)
	1 mark per mark point
	• function declaration taking appropriate parameters **and** recursive calls
	• remainder of the function is accurate including appropriate DIV operator.
	
	Example program code:
	VB.NET
	`Function BinarySearch(ByVal SearchArray(,) As Integer, Lower As Integer, Upper As Integer, SearchValue As Integer)`
	`  Dim Mid As Integer`
	`  If Upper >= 0 Then`
	`  Mid = (Lower + (Upper - 1)) \ 2`
	`  If SearchArray(0, Mid) = SearchValue Then`
	`  Return Mid`
	`  ElseIf SearchArray(0, Mid) > SearchValue Then`
	`  Return BinarySearch(SearchArray, Lower, Mid - 1, SearchValue)`
	`  Else`
	`  Return BinarySearch(SearchArray, Mid + 1, Upper, SearchValue)`
	`  End If`
	`  End If`
	`  Return -1`
	`End Function`
	
	Python
	`def BinarySearch(SearchArray, Lower, Upper, SearchValue):`
	`  if Upper >= 0:`
	`  Mid = int((Lower + (Upper - 1)) / 2)`
	`  If SearchArray[0][Mid] == SearchValue:`
	`  return Mid`
	`  elif SearchArray[0][Mid] > SearchValue:`
	`  return BinarySearch(SearchArray, Lower, Mid-1, SearchValue)`
	`  else:`
	`  return BinarySearch(SearchArray, Mid+1, Upper, SearchValue)`
	`  return -1`
		8

Question	Answer	Marks

2(c)(i)	**Java**
	`public static Integer BinarySearch(Integer[][] SearchArray, Integer Lower, Integer Upper, Integer SearchValue){`
	`  Integer Mid = 0;`
	`  If Upper >= 0 {`
	`  Mid = (Lower + (Upper - 1)) / 2;`
	`  If SearchArray[0][Mid] == SearchValue ){`
	`  return Mid;`
	`  }else if SearchArray[0][Mid] > SearchValue {`
	`  return BinarySearch(SearchArray, Lower, Mid-1, SearchValue);`
	`  }else{`
	`  return BinarySearch(SearchArray, Mid+1, Upper, SearchValue);`
	`  }`
	`  } return -1;`
	`  }`	

Question	Answer	Marks
2(c)(ii)	1 mark per mark point • screenshot outputting the index when Number is found • screenshot outputting –1 with a Number not found e.g.  	**2**

Question	Answer	Marks

3(a)	1 mark per mark point
	• `Card` class declaration (and end where appropriate)
	
	• Both attributes (`Number` and `Colour`) declared with suitable data types …
	• …as private
	
	• correct constructor header (and end where appropriate) with **two** parameters …
	• …both **parameters** assigned to the attributes
	
	Example program code:
	VB.NET
	`Class Card`
	`  Private Number As Integer`
	`  Private Colour As String`
	`  Sub New(Numberp, Colourp)`
	`  Number = Numberp`
	`  Colour = Colourp`
	`  End Sub`
	`End Class`
	
	Python
	`class Card:`
	`  #Number as Integer`
	`  #Colour as string`
	`  def __init__(self, Numberp, Colourp):`
	`  self.__Number = Numberp`
	`  self.__Colour = Colourp`
		5

Question	Answer	Marks

3(a)	**Java**
	`import java.util.Scanner;`
	`import java.io.*;`
	`class Card{`
	`  private Integer Number;`
	`  private String Colour;`
	`  public Card(Integer pNumber, String pColour){`
	`  Number = pNumber;`
	`  Colour = pColour;`
	`  }`
	`  public static void main(String args[]){`
	`  }`
	`}`	

Question	Answer	Marks

3(b)	1 mark per mark point
	• 1 get method header (and close where appropriate) with no parameter …
	• … returning attribute
	
	• 2nd correct get method
	
	Example program code:
	VB.NET
	`Function GetNumber()`
	`  Return Number`
	`End Function`
	`Function GetColour()`
	`  Return Colour`
	`End Function`
	
	Python
	`def GetNumber(self):`
	`  return self.__Number`
	`def GetColour(self):`
	`  return self.__Colour`
	
	Java
	`public Integer GetNumber(){`
	`  return Number;`
	`}`
	
	`public String GetColour(){`
	`  return Colour;`
	`  }`
		3

Question	Answer	Marks

3(c)	1 mark per mark point to max 7
	• Declaration of array with 30 elements of type `Card`
	
	• Opening the text file `CardValues.txt` for read
	• Looping until EOF/30 times
	
	• Reading in **all** sets of **2** lines (number then colour) …
	• …creating object of type `Card`…
	• …with number and colour read in from file …
	• …storing in next array element
	
	• Try and catch for file handling…s
	• ….with appropriate outputs
	
	• Closing the file in a **suitable** place
	
	Example program code:
	VB.NET
	`Sub Main()`
	`  Dim CardArray(0 To 29) As Card`
	`  Dim NumberRead As Integer`
	`  Dim ColourRead As String`
	
	`  Try`
	`  Dim Filename As String = "CardValues.txt"`
	`  Dim FileReader As New System.IO.StreamReader(filename)`
	`    For x = 0 To 29`
	`      NumberRead = FileReader.ReadLine()`
	`      ColourRead = FileReader.ReadLine()`
	`      CardArray(x) = New Card(NumberRead, ColourRead)`
	`    Next`
	`    FileReader.close()`
	`    Catch ex As Exception`
		7

Question	Answer	Marks

3(c)	Console.WriteLine("Invalid file")
	End Try
	End Sub
	
	Python
	CardArray = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] #integer
	
	try:
	Filename = "CardValues.txt"
	File = open(Filename,'r')
	for x in range(0,30):
	NumberRead = int(File.readline())
	ColourRead = File.readline()
	CardArray[x] = Card(NumberRead, ColourRead)
	
	File.close
	except IOError:
	print("Could not find file")
	
	Java
	public static void main(String args[]){
	Card[] CardArray = new Card[30];
	Integer NumberRead;
	String ColourRead;
	String FileName = "CardValues.txt";
	try{
	FileReader F = new FileReader(FileName);
	BufferedReader Reader = new BufferedReader(f);
	for(Integer x = 0; x < 30; x++){
	NumberRead = Integer.parseInt(Reader.readLine());
	ColourRead = Reader.readLine();
	CardArray[x] = new Card(NumberRead, ColourRead);
	}
	Reader.close();
	}	

Question	Answer	Marks

3(c)	`  catch(FileNotFoundException ex){`
	`  System.out.println("No file found");`
	`  }`
	`  catch(IOException ex){`
	`  System.out.println("No file found");`
	`  }`
	`}`	

Question	Answer	Marks

3(d)	1 mark per mark point to max 6
	• Implementing a suitable way of storing which card have been selected
	
	• Function `ChooseCard()` header (and close) **and** returning an integer (index) in all cases
	
	• Reading in array index from the user …
	• …with suitable validation **looping** until it is between 1 and 30 (inclusive)
	
	• Converting input to array index (e.g. –1 each time)
	
	• Check if the input is already selected…
	• … if it is selected, loop until index input is not already selected
	• … returning index of available card selected
	
	• Stores the valid Card chosen as taken (using any suitable method)
	
	Example program code:
	VB.NET
	`Dim NumbersChosen(0 To 29) As Boolean`
	`Sub Main()`
	`  For x = 0 To 29`
	`  NumbersChosen(x) = False`
	`  Next`
	`….`
	`End Sub`
	`Function chooseCard()`
	`  Dim CardSelected As Integer`
	`  Dim flagContinue As Boolean = True`
	`  While flagContinue = True`
	`  Console.WriteLine("Select a Card from 1 to 30")`
	`  CardSelected = Console.ReadLine()`
	`  If CardSelected < 1 Or CardSelected > 30 Then`
	`  Console.WriteLine("Number must be between ")`
		6

Question	Answer	Marks

3(d)	ElseIf NumbersChosen(CardSelected - 1) = True Then
	Console.WriteLine("Already taken")
	Else
	Console.WriteLine("valid")
	flagContinue = False
	End If
	End While
	NumbersChosen(CardSelected - 1) = True
	Return CardSelected - 1
	End Function
	
	Python
	global NumbersChosen
	…
	def chooseCard ():
	global NumbersChosen
	flagContinue = True
	while flagContinue == true:
	CardSelected = int(input("Select a Card from 1 to 30"))
	if CardSelected < 1 or CardSelected > 30:
	print("Number must be between 1 and 30")
	elif NumbersChosen(CardSelected - 1) == True:
	print("Already taken")
	else:
	print("Valid")
	flagContinue = False
	NumbersChosen[CardSelected-1] = True
	return CardSelected-1
	…
	#main
	…
	NumbersChosen = [False for i in range(30)]	

Question	Answer	Marks

3(d)	**Java**
	`public static Boolean[] NumbersChosen = new Boolean[30];`
	
	`public Integer chooseCard (){`
	`Boolean flagContinue = true;`
	`Integer CardSelected = -1;`
	`while(flagContinue){`
	`System.out.println("Select a Card from 1 to 30");`
	`Scanner scanner = new Scanner(System.in);`
	`CardSelected = Integer.parseInt(scanner.nextLine());`
	`if(CardSelected < 1 || CardSelected > 30){`
	`System.out.println("Number must be between 1 and 30");`
	
	`}else if(NumbersChosen[CardSelected - 1]){`
	`System.out.println("Already taken");`
	`}else{`
	`System.out.println("Valid");`
	`flagContinue = false;`
	`}`
	`}`
	`NumbersChosen[CardSelected - 1] = true;`
	`return CardSelected - 1;`
	`}`	

Question	Answer	Marks

3(e)(i)	1 mark per mark point
	• declaring array `Player1` of type `Card`
	• calling the function `ChooseCard()` **four** times
	• storing the **card**, that is in the index returned, in the array `Player1`
	
	• outputting all **four** numbers **and** colours in `Player1` …
	• …. using the get methodss
	
	Example program code:
	VB.NET
	`Dim Player1(0 To 3) As Card`
	`For x = 0 To 3`
	`  Player1(x) = CardArray(ChooseCard(NumbersChosen))`
	`Next`
	`for x = 0 to 3`
	`  console.writeline(Player1(x).GetColour)`
	`  console.writeline(Player1(x).GetNumber)`
	`next x`
	
	Python
	`Player1 = [] #of type Card`
	`for x in range(0, 4):`
	`    ReturnNumber = ChooseCard ()`
	`    Player1.append(CardArray[ReturnNumber])`
	`for x in range(0, 4):`
	`    print(Player1[x].GetColour())`
	`    print(Player1[x].GetNumber())`
		5

Question	Answer	Marks

3(e)(i)	**Java**
	`Card[] Player1 = new Card[5];`
	`  for(Integer x = 0; x < 5; x++){`
	`  Player1[x] = CardArray[ChooseCard()];`
	`  }`
	`  for(Integer x = 0; x < 5; x++){`
	`  System.out.println(Player1[x].GetColour());`
	`  System.out.println(Player1[x].GetNumber());`
	`  }`	

3(e)(ii)	1 mark for both tests
	Test 1: inputting 1, 5, 9, 10. Outputting: 1 red 9 green 9 orange 10 red
	Test 2: inputting 2 2 3 4 4 5. Outputting: 2 already taken. Then 5 black 2 while 4 red 9 green
	Test 1 e.g.
	`Select a card from 1 to 301`
	`Valid`
	`Select a card from 1 to 305`
	`Valid`
	`Select a card from 1 to 309`
	`Valid`
	`Select a card from 1 to 3010`
	`Valid`
	`1`
	`red`
	`9`
	`green`
	`9`
	`orange`
	`10`
	`red`
		1

Question	Answer	Marks
3(e)(ii)	Test 2 e.g. Select a card from 1 to 302 Valid Select a card from 1 to 302 Already taken Select a card from 1 to 303 Valid Select a card from 1 to 304 Valid Select a card from 1 to 304 Already taken Select a card from 1 to 305 Valid 5 black 2 white 4 red 9 green	