N08/5/COMSC/HP2/ENG/TZ0/XX/M+

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

November 2008

COMPUTER SCIENCE

Higher Level

Paper 2

14 pages

Subject Details: Computer Science HL Paper 2 Markscheme

Mark Allocation

Candidates are required to answer ALL questions ***[20 marks]*** for question 1, ***[20 marks]*** for question 2, ***[20 marks]*** for question 3 and ***[40 marks]*** for question 4. Maximum total = ***[100 marks]***.

General

A markscheme often has more specific points worthy of a mark than the total allows. This is intentional. Do not award more than the maximum marks allowed for that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

- Each statement worth one point has a separate line and the end is signified by means of a semi-colon (;).

- An alternative answer or wording is indicated in the markscheme by a "/"; either wording can be accepted.

- Words in (…) in the markscheme are not necessary to gain the mark.

- If the candidate's answer has the same meaning or can be clearly interpreted as being the same as that in the markscheme then award the mark.

- Mark positively. Give candidates credit for what they have achieved, and for what they have got correct, rather than penalising them for what they have not achieved or what they have got wrong.

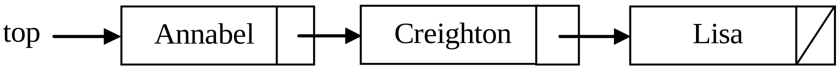
- Remember that many candidates are writing in a second language; be forgiving of minor linguistic slips. In this subject effective communication is more important than grammatical accuracy.

- Occasionally, a part of a question may require a calculation whose answer is required for subsequent parts. If an error is made in the first part then it should be penalized. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with **"FT"**.

1. (a) A call to the constructor (in class `Node`) is made;
This allocates storage space for the Node data;
Both data members are set to null; ***[3 marks]***

(b) *Award marks as follows up to* ***[3 marks max]****.*
Award ***[1 mark]*** *for top pointing to first node.*
Award ***[1 mark]*** *for names in the correct order.*
Award ***[1 mark]*** *for null pointer at end.*
Award ***[1 mark]*** *for arrows connecting each node.*

Example:



(Award ***[3 marks]*** *for correct diagram, arranged vertically.)* ***[3 marks]***

(c) *Award marks as follows up to* ***[7 marks max]****.*
Award ***[1 mark]*** *for specifying `pop` is a String method.*
Award ***[1 mark]*** *for testing for empty stack.*
Award ***[1 mark]*** *for calling `getName()` method correctly.*
Award ***[1 mark]*** *for setting `top` to next node (allow `top = top.next`).*
Award ***[1 mark]*** *for saving the name* **before** *advancing `top`.*
Award ***[1 mark]*** *for adjusting `top` using `getNext()` method.*
Award ***[1 mark]*** *for error message on empty stack.*
Award ***[1 mark]*** *for returning an appropriate String (i.e. correct name or empty String).*

Example:

```
public String pop()
{
  String name = "";
  if (top != null)
  {
    name = top.getName();
    top = top.getNext();
  }
  else
  {
    output("Error, nothing to pop");
  }
  return name;
}
```

[7 marks]

continued...

Question 1 continued

(d) *Award marks as follows up to **[3 marks max]**.*
*Award **[1 mark]** for loop through original array.*
*Award **[1 mark]** for correct use of `push()` (allow `push()` on its own).*
*Award **[1 mark]** for correct use of `pop()` (allow `pop()` on its own).*

Candidates may also use the condition `x < names.length` in the loops.

Example:

```
for (int x = 0; x < 6; x++)
{
  s.push(names[x]);
}
for (int x = 0; x < 6; x++)
{
  names[x] = s.pop();
}
```

[3 marks]

(e) *Award **[1 mark]** for a method and **[1 mark]** for a description up to **[2 marks]** × 2 = **[4 marks max]**.*

`isEmpty();`	returns true if the stack is empty;
`top();`	returns the top value without removing it;
`isFull();`	returns true if the stack is full;
`size();`	returns the number of items in the stack;

[4 marks]

Total: [20 marks]

2. (a) *Award marks as follows up to **[6 marks max]**.*
*Award **[1 mark]** for correct initialisation of local identifiers.*
*Award **[1 mark]** for loop until found.*
*Award **[1 mark]** for loop until top exceeded or equalled.*
*Award **[1 mark]** for incrementing `pos` or equivalent.*
*Award **[1 mark]** for return of correct value for `pos` if found.*
*Award **[1 mark]** for return of correct value for `pos` if word not found.*

Allow a test for String equality such as: `if (word == words[pos])`.
Do not penalize for clumsy or inelegant code (e.g. `if (found == false)`).

Example 1:

```
public int getPosition(String[] words, String word, int last)
{
  boolean found = false;
  int pos = 0;
  while ( (!found) && (pos <= last) )
  {
    found = word.equals(words[pos]);
    if (!found)
    {
      pos = pos+1;
    }
  }
  if (!found)
  {
    return (last+1);
  }
  else
  {
    return pos;
  }
}
```

Example 2:

```
public static int getPosition(String[] words, String word, int last)
{
  for (int i = 0; i <=last; i++)
  {
    if (word.equals(words[i])) return i;
  }
  return (last+1);
}
```

[6 marks]

(b) 255; ***[1 mark]***

continued...

Question 2 continued

(c) *Award **[1 mark]** for a clear advantage and **[1 mark]** for a clear disadvantage.
Award **[1 mark]** for **each** elaboration, up to **[4 marks max]**.*

Examples:
Advantages
The programming is less complex;
And therefore less liable to error;
The access time is likely to be faster;
As it is direct rather than sequential;
(Possibly a dubious statement for this example, but accept anyway.)
Disadvantages
The size of the array is fixed;
And therefore the stack might run out of space;
The linked list can grow as needed; ***[4 marks]***

(d) *Award marks as follows up to **[9 marks max]**.
Award **[1 mark]** for correct, revised, parameter list.
Award **[1 mark]** for test for `bottom > top` or equivalent.
Award **[1 mark]** for return of `top + 1` if true.
Award **[1 mark]** for test for item found.
Award **[1 mark]** for return of `mid` if true.
Award **[1 mark]** for comparison of `word` and word at middle of list.
Award **[1 mark]** for correct call to upper part.
Award **[1 mark]** for correct call to lower part.
(Award **[1 mark]** for worthy but incorrect attempt at both.)
Award **[1 mark]** for appropriate use of return statement in calls to upper and lower (or saving in identifier to return at end).*

Example 1:

```
public int getPosition(String[] list, String word, int bottom, int top)
{  if (bottom > top)
   {  return (top + 1);
   }
   else
   {  int mid = (bottom + top)/2;
      if (list[mid].equals(word))
      {  return  mid;
      }
      else if(word.compareTo(list[mid])< 0)
      {  // search lower part
         return getPosition(list, word, bottom, mid - 1);
      }
      else
      {  // search upper part
         return getPosition(list, word, mid + 1, top);
      }
   }
}
```

***Note:** This solution assumes that the word will always be in the list.*

continued...

Question 2 (d) continued

Example 2:

```
public static int getPosition(String[] words, String word, int last)
{
  int res = getPosition_aux(words, word, 0, (int)last);
  if (res == -1) return (last+1);
  else return res;
}

public static int getPosition_aux(String[] words, String word, int
                                  first, int last)
{
  if (first > last) return -1;
  else
  {
    int mid = (first + last) / 2
    int res = word.compareTo(words[mid]);
    if (res == 0) return mid;
    if (res < 0) return getPosition_aux(words, word, first, mid-1);
    else return getPosition_aux(words, word, mid+1, last);
  }
}
```

[9 marks]

Note: *This solution returns* `(last+1)` *if the value is not found.*
Award a further ***[2 marks]*** *for including this condition and any suitable return value* `(-1, last+1)`*; . The total of* ***[9 marks]*** *must not be exceeded.*

Total: [20 marks]

3. (a) (i) *Award up to **[4 marks max]**.*
The photographs need not be ordered/re-ordered (in their file);
When insertions are made to the file;
As the second file can contain a record number (as a field);
The second file can also be fully indexed;
To allow searching retrieval by other fields; ***[4 marks]***

(ii) *Award up to **[4 marks max]**.*
The photographs make the record size very large;
When insertions are made the order of records may change;
Which will be (comparatively) slow;
However, this file could also be indexed/fully indexed;
And insertion would only require a change in the index;
Not the main file; ***[4 marks]***

(b) *Award up to **[4 marks max]**.*
The file can be split into smaller units;
Which could be stored in main memory;
Each unit could then be sorted;
And saved as a file;
The resulting files could then be merged back;
As merging does not require the whole file to be in memory; ***[4 marks]***

(c) 2004; ***[1 mark]***

(d) O(1); ***[1 mark]***

(e) *Award **[1 mark]** for identifying a problem (accept if implicit), **[1 mark]** for describing it further and **[1 mark]** for identifying a solution.*
*Award up to **[3 marks max]** × 2 = **[6 marks max]**.*

Problem: Table access range
The whole table cannot be accessed;
For example, records will all be in the 2000-2008 range;
Include the time, for example, minutes/seconds;
Reverse the date format to yyyymmdd;
(Any reasonable suggestion).
Problem: Inadequate hash function
Function does not produce a large spread of numbers;
Select a better function, for example multiply day, month and year together and choose 4-digits from the middle; *(any reasonable suggestion)*
This will lead to fewer clashes/clustering of values;
Problem: Table size
The table size of 10,000 is inefficient;
Since the table will be always nearly full;
A larger hash table is more efficient/fewer clashes will occur;
An ideal size will be of the order of 12,000 – 14,000 entries;
(Any reasonable value). ***[6 marks]***

Total: [20 marks]

4. (a) *Award **[1 mark]** for a way (method and appropriate disability) and a further **[1 mark]** for some elaboration up to **[2 marks max]** × 3 = **[6 marks max]**.*
Accept any reasonable but distinct examples.

Screen readers for partially sighted or blind people;
Can be used to read web pages or other text documents;
Can be used to read books/documents with a scanner;

Screen readers for people with touch/dexterity problems;
Can be used to navigate through web pages or other text documents (without having to use keyboard or mouse);

Alert systems for partially sighted or blind people;
Can be used to indicate arrival of email (by reading pop up messages);
Can be used to alert them to errors (by reading pop up error messages);
etc. ***[6 marks]***

(b) (i) *Award up to **[4 marks max]**.*
Using macros;
To provide shortcut keys for lengthy items (*e.g.* addresses of friends);
Using built-in features of software;
Such as auto-replace/correct functions;
Adding sound to user interfaces;
Such as providing audible feedback on key presses; ***[4 marks]***

(ii) *Award up to **[4 marks max]**.*
A screen magnification system helps users see what they have written;
When the user places the cursor over text it is enlarged;
And/or shown in a high contrast font;
This can be used with a large monitor;
As enlarged text takes up more space;

A speech input system helps users enter text;
Without having to use a keyboard;
The user needs a microphone (as an input device);
And speech is converted to text; ***[4 marks]***

Note: *Only accept answers relating to software.*

continued…

Question 4 continued

(c) *Award* ***[1 mark]*** *for an issue and* ***[1 mark]*** *for an elaboration, up to* ***[2 marks max]*** *× 2 =* ***[4 marks max]****.*

Avoid using low contrast or hard to see colours;
Which may affect colour blind users;

Avoid having pages with excessive scrolling;
Which make it difficult for users with dexterity issues;

Provide alt tags for images;
For users using screen readers;

Avoid the use of fancy fonts/very small fonts;
Which can be hard to read for visually impaired;

Avoid the use of sounds with no visual cues;
Which won't be noticed by hearing impaired users;
etc. ***[4 marks]***

(d) (i) *Award* ***[1 mark]*** *for a suggestion and* ***[1 mark]*** *for an elaboration, up to* ***[2 marks max]*** *× 2 =* ***[4 marks max]****.*

Using keyboards modified to reduce the need for hand movement;
Such as grouping keys together;

Placing more commonly used keys near the centre of the keyboard;
Providing audible feedback on key presses;

Using a replacement for the mouse;
Such as a large trackball which does not require fine hand movement;

Provide cordless, optical mice;
Which are lower maintenance and easier to move;
etc. ***[4 marks]***

(ii) *Award* ***[1 mark]*** *for a suggestion and* ***[1 mark]*** *for an elaboration, up to* ***[2 marks max]*** *× 2 =* ***[4 marks max]****.*

Using keyboards with electronic Braille displays;
So that these users can still "read" the screen;

Speech output system;
To verify the data that is being printed;

Text-to-speech system;
In case printed documents such as schedules and tickets need to be read;
etc. ***[4 marks]***

Note: *Only accept answers relating to hardware.*

continued…

Question 4 continued

(e) *Award up to **[5 marks max]**. Accept any reasonable and feasible suggestion.*
A head-mounted pointing device;
Could be used in place of a mouse;
This can be used with a keyboard emulator;
Which works by placing a (virtual) keyboard on screen;
Enabling the user to type messages;
or
Speech recognition/voice synthesis (accept “speech system”) could be used;
For example, to access hyperlinks;
And other controls on a page;
Also to enable “chat as text” to be converted to the spoken word;
And (equally) spoken words to appear as text in a chat room; ***[5 marks]***

(f) *Award **[1 mark]** for a consequence and **[2 marks]** for a good discussion. (Award **[1 mark]** only for an elaboration with some credit.)*
*Award up to **[3 marks max]** × 3 = **[9 marks max]**.*
(The consequence may be implicit – award the mark if it is.)

Equality of opportunity;
Disabled users compete in the marketplace for jobs;
They may be disadvantaged because they cannot use a computer with full efficiency;

Quality of life;
Many non-disabled users have friends whom they communicate with;
Disabled persons may be disadvantaged because they cannot access these tools, (or only partially);

Ease of access;
Much business, public and private is carried out by or with computers;
This puts disabled people at a disadvantage if they cannot access software/hardware that has been designed for them;

Improved access due to technologies developed elsewhere;
May improve opportunity/quality of life/ease of access;
Compared to division on other grounds;
Such as race/gender/income; ***[9 marks]***

Note: *Accept suitable alternative answers.*

Total: [40 marks]