N11/5/COMSC/HP2/ENG/TZ0/XX/M

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

MARKSCHEME

November 2011

COMPUTER SCIENCE

Higher Level

Paper 2

12 pages

General Marking Instructions

After marking a sufficient number of scripts to become familiar with the markscheme and candidates' responses to all or the majority of questions, Assistant Examiners (AEs) will be contacted by their Team Leader (TL). The purpose of this contact is to discuss the standard of marking, the interpretation of the markscheme and any difficulties with particular questions. It may be necessary to review your initial marking after contacting your TL. **DO NOT BEGIN THE FINAL MARKING OF YOUR SCRIPTS IN RED INK UNTIL YOU RECEIVE NOTIFICATION THAT THE MARKSCHEME IS FINALIZED.** *You will be informed by e-mail, fax or post of modifications to the markscheme and should receive these about one week after the date of the examination. If you have not received them within 10 days you should contact your TL and IB Cardiff. Make an allowance for any difference in time zone before calling.* **AEs WHO DO NOT COMPLY WITH THESE INSTRUCTIONS MAY NOT BE INVITED TO MARK IN FUTURE SESSIONS.**

You should contact the TL whose name appears on your "Allocation of Schools listing" sheet.

Note:
Please use a personal courier service when sending sample materials to TLs unless postal services can be guaranteed. Record the costs on your examiner claim form.

General Marking Instructions

1. Once markscheme is received mark in pencil until final markscheme is received.

2. Follow the markscheme provided, do **not** use decimals or fractions and mark only in **RED**.

3. Where a mark is awarded, a tick (✓) should be placed in the text at the **precise point** where it becomes clear that the candidate deserves the mark.

4. Sometimes, careful consideration is required to decide whether or not to award a mark. Indeed, another examiner may have arrived at the opposite decision. In these cases write a brief annotation in the **left hand margin** to explain your decision. You are encouraged to write comments where it helps clarity, especially for moderation and re-marking.

5. Unexplained symbols or personal codes/notations on their own are unacceptable.

6. Record subtotals (where applicable) in the right-hand margin against the part of the answer to which they refer. Show a mark for each part question (a), (b), *etc.* Do **not** circle sub-totals. Circle the total mark for the question in the right-hand margin opposite the last line of the answer.

7. Where an answer to a part question is worth no marks, put a zero in the right-hand margin.

8. Record the mark awarded for each of the four questions answered in the Examiner Column on the cover sheet. Add up the marks awarded and enter this in the box marked TOTAL in the Examiner Column on the cover sheet.

9. After entering the marks on the cover sheet check your addition of all marks to ensure that you have not made an arithmetical error. Check also that you have transferred the marks correctly to the cover sheet. **We have script checking and a note of all clerical errors may be given in feedback to all examiners.**

10. Every page and every question must have an indication that you have marked it. Do this by **writing your initials** on each page where you have made no other mark.

11. A candidate can be penalized if he/she clearly contradicts him/herself within an answer. Once again make a comment to this effect in the left hand margin.

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) Insert a class data item to link `Animal` objects together,
`public Animal nextAnimal;` ***[1 mark]***

(b) *Award marks as follows up to* ***[3 marks max]****.*
Award ***[1 mark]*** *for describing linking the* `Animal` *objects sequentially in a list;*
Award ***[1 mark]*** *for setting* `nextAnimal` *to* `null` *in the last* `Animal` *object to mark the list's end;*
Award ***[1 mark]*** *for a correct diagram;* ***[3 marks]***

(c) *Award marks as follows up to* ***[4 marks max]****.*
Award ***[1 mark]*** *for correctly looping through the species in* `allSpecies`*;*
Award ***[1 mark]*** *for correctly testing the names using the string 'equals' method;*
Award ***[1 mark]*** *for correctly determining the index for the species of each animal;*
Award ***[1 mark]*** *for correct return value;*

Example answer:

```
public int getSpeciesIndex(String species)
{
  int iSpecies = 0;
  while (!allSpecies[iSpecies].equals(species)
         && iSpecies < 6)   // go through each species
  {
    iSpecies = iSpecies + 1;
  }
  if (iSpecies < 6) return iSpecies;
  else return -1;
}
```

[4 marks]

(d) *Award marks as follows up to* ***[4 marks max]****.*
Award ***[1 mark]*** *for use of loop to traverse linked list;*
Award ***[1 mark]*** *for correct loop termination when* `null` *is encountered;*
Award ***[1 mark]*** *for correct counting of animals;*
Award ***[1 mark]*** *for correct return value;*

Example answer:

```
int speciesCount(String species)
{
  // get the index for the requested species
  int iSpecies = getSpeciesIndex(species);

  // count and print animals in the appropriate list
  int iCount = 0;
  Animal beast = speciesList[iSpecies];
  while (beast != null)
  {
    iCount = iCount + 1;
    beast = beast.nextAnimal;
  }
  return iCount;
}
```

[4 marks]

continued …

Question 1 continued

(e) *Award marks as follows up to* ***[8 marks max]****.*
(For each of the following, award ***[1 mark]*** *for an attempt or* ***[2 marks]*** *for a correct implementation.)*
Award ***[2 marks]*** *for looping through all the animals in* `animalList`*;*
Award ***[2 marks]*** *for correctly inserting each animal into a list;*
Award ***[2 marks]*** *for inserting animals into the correct list (determining which list to insert into);*
Award ***[2 marks]*** *for* ***either*** *leaving* `animalList` *empty* ***or*** *cloning the* `Animal` *objects and leaving the list* `animalList` *intact;*

Example answer:

```
public void divideAnimals()
{
  Animal temp = animalList;
  while (temp != null)
  {
    int index = getSpeciesIndex(temp.animalSpecies);
    if (speciesList[index] == null)
    {
      speciesList[index] = new Animal(temp.animalID, temp.animalSpecies)
    }
    else
    {
      Animal spare = new Animal(temp.animalID, temp.animalSpecies);
      spare.next = speciesList[index];
      speciesList[index] = spare;
    }
    temp = temp.next;
  }
}
```

The above solution preserves the original `animalList` *linked list by cloning each* `animalList` *object.*

Alternatively, each `animalList` *object could be deleted from this list as it is added to the new list.* ***[8 marks]***

Total: [20 marks]

2. (a) The method `FishCounts()` is a constructor method;
It allows new `FishCounts` objects to be initialized;
The method initializes all the `counts` elements to `-1`; ***[3 marks]***

(b) *Award marks as follows up to* ***[4 marks max]****.*
Award ***[1 mark]*** *for starting the loop with second element;*
Award ***[1 mark]*** *for ending the loop with penultimate element;*
Award ***[1 mark]*** *for correctly identifying elements needing to be filled-in;*
Award ***[1 mark]*** *for correctly computing the mean;*

Example answer:

```
public void fillMissingCounts()
{
  for (int i = 1; i < 23; i = i + 1)
  {
    if (counts[i] < 0 )
      counts[i] = (counts[i - 1] + counts[i + 1]) / 2;
  }
}
```

[4 marks]

(c) Calculating the mean requires doing division;
Representing the result as an **`int`** can cause truncation;
A suitable example such as `3 / 2 = 1.5` will set `mean = 1`; ***[3 marks]***

continued …

Question 2 continued

(d) *Award marks as follows up to **[10 marks max]**.*
*Award **[1 mark]** for initializing sum for* `mean` *to zero;*
*Award **[1 mark]** for initializing* `max` *to zero or a negative value;*
*Award **[1 mark]** for correctly looping through all elements of* `countData`*;*
*Award **[1 mark]** for correctly finding the maximum count;*
*Award **[1 mark]** for correctly updating* `statsObj.max` *and* `statsObj.maxIndex`*;*
*Award **[1 mark]** for correctly computing the mean;*
*Award **[1 mark]** for correctly updating* `statsObj.mean`*;*
*Award **[1 mark]** for correctly excluding the maximum count from the search for the second-largest;*
*Award **[1 mark]** for correctly finding the second-largest count;*
*Award **[1 mark]** for correctly updating* `statsObj.max2`*;*

Example answer:

```
public FishStats findTopTwo()
{
  FishStats statsObj = new FishStats();

  // find largest count and mean
  statsObj.max = 0;
  statsObj.mean = 0;
  for (int i = 0; i < 24; i = i + 1)
  {
    if (counts[i] > statsObj.max)
    {
      statsObj.max = counts[i];
      statsObj.maxIndex = i;
    }
    statsObj.mean = statsObj.mean + counts[i];
  }
  statsObj.mean = statsObj.mean / 24;

  // find the second-largest count
  statsObj.max2 = 0;
  for (int i = 0; i < 24; i = i + 1)
  {
    if ((i != statsObj.maxIndex) && (counts[i] > statsObj.max2))
      statsObj.max2 = counts[i];
  }
  return statsObj;
}
```

[10 marks]

Total: [20 marks]

3. (a) *Award **[1 mark]** for the idea of a record being a group of data items, or fields;*
*Award **[1 mark]** for the idea that there is some sort of identifier;*
An aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. ***[2 marks]***

(b) (i) Sort the file into alphabetical order by name;
Select every nth record;
Save name and offset for each selected record; ***[3 marks]***

(ii) Sequentially search the partial-index for the name prior to the one sought;
Access the `refugees` file at that point and search sequentially from there; ***[2 marks]***

(c) *Award up to **[3 marks max]**.*
Each time a record is added, the file must be re-sorted and re-indexed;
The file records must be of a fixed, uniform size;
Each time a record is removed the index must be rebuilt;
The file must be stored in a random access medium; ***[3 marks]***

(d) *Award up to **[3 marks max]**.*
The key field must be unique for each record;
It cannot simply be the refugee name;
Could be a unique code number that includes the camp code so independent camps cannot assign the same code; ***[3 marks]***

(e) Records changed in the camp's refugee file (inserts, edits and deletions);
File sorted by key field; ***[2 marks]***

(f) Read a record from transaction file;
Read records from master file until matching record is located or not found;
Insert or update record, as needed;
Repeat until all the records in the transaction file have been processed;
Repeat until all the transaction files have been processed; ***[5 marks]***

Total: [20 marks]

4. (a) (i) *Award* ***[1 mark]*** *for a simple statement, award an additional* ***[1 mark]*** *for a more detailed explanation.*
Running the new and old systems in parallel; ***[2 marks]***

(ii) *Award up to* ***[2 marks max]*** *for simply stating a reason, award up to an additional* ***[2 marks]*** *for more complete and detailed explanations.*
To save money;
To avoid rebuilding the machines;
To be able to test software before the building was done; ***[4 marks]***

(iii) *Award up to* ***[2 marks max]****.*
The consequences of system failure;
The time available;
The cost;
Whether all possibilities will be covered; ***[2 marks]***

(b) (i) *Award* ***[1 mark]*** *for identifying a suitable method and* ***[1 mark]*** *for a more complete outline.*

Example methods:
Wi-Fi;
Wired network access (the cart must plug in at gates/kiosks/*etc.*);
Some sort of proprietary radio network; ***[2 marks]***

(ii) *Award* ***[1 mark]*** *each for up to* ***two*** *potential problems, award up to an additional* ***[2 marks]*** *for more complete and detailed explanations.*
Wi-Fi dead spots;
The airport Wi-Fi network is public;
The cart operator might forget to plug in the data connector;
Proprietary radio network ties you to a particular vendor;
Unauthorized users might plug into the data ports or use the Wi-Fi; ***[4 marks]***

(iii) *Award* ***[1 mark]*** *each for up to* ***two*** *ways to secure the communications. Award up to an additional* ***[2 marks]*** *for more complete and detailed descriptions.*
Encrypt the data;
Use a secure radio network;
Use a private Wi-Fi network;
Validate each cart’s transmissions with a challenge/response protocol; ***[4 marks]***

(c) (i) *Award* ***[1 mark]*** *each for up to* ***two*** *features identified in the case study, up to* ***[2 marks max]****.*
The interface is familiar;
The interface is easy-to-use;
Controllers can instantly understand the state of the traffic;
Controllers can communicate without interrupting each other;
Controllers know they have a new aircraft when they hear the printer; ***[2 marks]***

continued …

Question 4 continued

(ii) *Award* ***[1 mark]*** *each for up to* ***two*** *reasonable suggested ways, award an additional* ***[1 mark]*** *each for more complete descriptions of the suggestions.*
Use a touch screen to let controllers move shapes around the display;
Use an audible signal to indicate a new aircraft;
Allow controllers to drag and drop aircraft information so they drop it onto another controller's display; ***[4 marks]***

(iii) *Award up to* ***[6 marks max]****. Answer should include discussion of three distinct areas. Award up to* ***[2 marks max]*** *for the discussion in each area. Award* ***[1 mark]*** *if an area is simply mentioned, award an additional* ***[1 mark]*** *for a fuller discussion.*

Technical advantages/disadvantages of replacing paper strips, such as:
Integration with other systems;
Ability to work when computers fail;
Availability of records for incident investigations;

Social issues related to changing to computer-based system, such as:
Controllers like paper strips;
Training/re-training requirements;

The cost and likelihood of failures. ***[6 marks]***

(d) (i) *Award up to* ***[2 marks max]****.*
Scan their passport optically;
Scan their passport RFID;
Scan their fingerprint;
Scan their iris;
Any reasonable method; ***[2 marks]***

(ii) *Award up to* ***[4 marks max]****. Award* ***[1 mark]*** *each for up to* ***two*** *ways to manage the queues, award an additional* ***[1 mark]*** *each for a detailed, clear discussion.*
Automate check-in so it does not require the passenger to be involved;
Get passengers checked-in before they arrive at the airport;
Identify passengers and use the computer system to direct them to an agent when it is their turn, freeing them from having to wait in a queue;
Have sufficient check-in counters and kiosks so that there is never any waiting; ***[4 marks]***

(iii) *For each of* ***two*** *privacy and/or ethical concerns, award* ***[1 mark]*** *for identification of a legitimate concern and* ***[1 mark]*** *for a full description of it*
Concerns could include:
People seeing you log-in your detail at self-service check-ins;
Hackers intercepting personal details if check-in at home; ***[4 marks]***

Total: [40 marks]