



MARKSCHEME

May 2011

COMPUTER SCIENCE

Standard Level

Paper 2

11 pages

*This markscheme is **confidential** and for the exclusive use of examiners in this examination session.*

*It is the property of the International Baccalaureate and must **not** be reproduced or distributed to any other person without the authorization of IB Cardiff.*

General Marking Instructions

Subject Details: **Computer Science SL Paper 2 Markscheme**

Mark Allocation

Candidates are required to answer ALL questions *[20 marks]* for question 1, *[20 marks]* for question 2 and *[30 marks]* for question 3. Maximum total = *[70 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 penalizing them for what they have not achieved or what they have got wrong.
- 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.

1. (a) 245; **[1 mark]**

- (b) *Award marks as follows up to [3 marks max].*
Award [1 mark] for correct loop (accept any logically equivalent condition, e.g. $i < 6$ and/or other loop construct);
*Award [1 mark] for **if** test;*
Award [1 mark] for correct assignment to `bestSales`;

Example answer:

```
int bestSales = 0;
for (int i = 0; i <= 5; i = i + 1)
{
    if (sales[i] >= bestSales)
        bestSales = sales[i];
}
output("Best sales = " + bestSales);
```

[3 marks]

- (c) *Award marks as follows up to [2 marks max].*
Award [1 mark] for declaration/initialization/return of local variables;
Award [1 mark] for correct assignment of `bestID = i`;

Example answer:

```
int bestID = 0;
int bestSales = 0;
for (int i = 0; i <= 5; i = i + 1)
{
    if (sales[i] >= bestSales)
    {
        bestSales = sales[i]; // not necessary
        bestID = i;
    }
}
output("Best salesperson's ID = " + bestID);
```

[2 marks]

- (d) *Accept both answers, up to [1 mark max].*

```
theYearSales[1][4];
360;
```

[1 mark]

- (e) *Award marks as follows up to [3 marks max].*
Award [1 mark] for variables' declarations/initialization;
Award [1 mark] for correct loop;
Award [1 mark] for correct assignment
Award [1 mark] for correct return;

Example answer:

```
public int totalSales(int id)
{
    int tot = 0;
    for (int m = 0; m <= 11; m = m + 1)
    { tot = tot + theYearSales[m][id]; }
    return tot;
}
```

[3 marks]

continued ...

Question 1 continued

- (f) Award marks as follows up to **[6 marks max]**.
Award **[1 mark]** for variables' declarations;
Award **[1 mark]** for correct loop;
Award **[1 mark]** for correct **if** command;
Award **[1 mark]** for correct assignment of *bestTotal*;
Award **[1 mark]** for correct assignment of *bestYearID = i*;
Award **[1 mark]** for correct return;

Example answer 1 (all variables are declared at the beginning):

```
public int bestSalesperson()
{
    int bestYearID;
    int bestTotal = 0;
    int currentBestTotal;
    for (int i = 0; i <= 5; i = i + 1)
    {
        currentBestTotal = totalSales(i);
        if (currentBestTotal >= bestTotal)
        {
            bestTotal = currentBestTotal;
            bestYearID = i;
        }
    }
    return bestYearID;
}
```

Example answer 2 (some variables are introduced within the loop):

```
public int bestSalesperson()
{
    int bestYearID = 0;
    int bestTotal = 0;
    for (int i = 0; i <= 5; i = i + 1)
    {
        int currentBestTotal = totalSales(i);
        if (currentBestTotal >= bestTotal)
        {
            bestTotal = currentBestTotal;
            bestYearID = i;
        }
    }
    return bestYearID;
}
```

[6 marks]

continued ...

Question 1 continued

(g) *Award up to [4 marks max].*

A (global) array `names` of salespersons' names is present;
And it is ordered on the ID;

Use `bestYearID`;

To access/return the position in the array: `names[bestYearID]`;

Alternative

A parallel Array of stype string;

stores Name

ID used to access the Name in the array

return `name[id]`

Accept all more/less detailed descriptions such as the following ones.

Example answer 1:

In `bestSalesman`:

A String **variable** `bestName` is declared (/initialized with " "),
`bestName = names[bestYearID]; ... ; return bestName;`

Example answer 2:

`bestName` variable not declared and **return** `names[bestYearID]`

[4 marks]

Total: [20 marks]

2. (a) *Award marks as follows up to [2 marks max].*
Award [1 mark] for mentioning whole numbers compared to decimal numbers or equivalent;
Award [1 mark] for some reasonable reference to doubles taking more space than integers e.g. 32Bits vs. 64 bits, 4bytes vs. 8bytes, or double take more space to represent than integers; **[2 marks]**
- (b) To create/instantiate new objects;
From outside the class;
- Accept non-technical descriptions that show the student knows that public allows the class to be used by other processes, award [1mark max].* **[2 marks]**
- (c) *Award up to [3 marks max].*
A variable `c1` is declared;
Variable is of type `City`;
Objects fields are instantiated;
Objects variables passed values;
The variable `c1` is set to reference the created object;
- Accept NON-technical descriptive answers where the student implies some understanding that a variable is created and values used, award [1 mark max].* **[3 marks]**
- (d) “Cardiff”; **[1 mark]**
- (e) (i) *Award up to [2 marks max].*
Declare/use an array of type `City`;
With 100 locations;
Or
Use an array (*award [1 mark] only with no data type or length*);
Hence: an array with 100 spots is awarded **[1 mark]**;
Or
Write data to a file (sequential or random);
To be read later for processing; **[2 marks]**

continued ...

Question 2 continued

- (ii) *Award marks as follows up to [4 marks max].
Award [1 mark] for declaration/initialization of a temporary object or variables;
Award [1 mark] for correct loop;
Award [1 mark] for correct conditional statement;
Award [1 mark] for correct return;*

Example answer 1 (using a temporary object):

```
// global declaration of cityArray
public String mostPopCity(City[] cityArray)
{
    City temp;
    temp = cityArray[0];
    for (int i = 1; i < cityArray.length; i = i + 1)
    {
        if (cityArray[i].population > temp.population)
        { temp = cityArray[i]; }
    }
    return temp.name;
}
```

Example answer 2 (using local variables):

```
public String mostPopCity(City[] cityArray)
{
    int max = 0;
    String name = "";
    for (int i = 0; i < cityArray.length; i = i + 1)
    {
        if (cityArray[i].population > max)
        {
            max = cityArray[i].population;
            name = cityArray[i].name;
        }
    }
    return name;
}
```

[4 marks]

continued ...

Question 2 continued

- (f) Do **NOT** accept anything that is clearly not a bubble sort.

Example Answer:

Starting from position 0 in the `cityArray/list`;
And for the whole length of the array/`list`;
Objects at consecutive indexes `i` and `i + 1` are compared for their field
population;
And if the value of object `i` is bigger than the value of object `i + 1` the two
objects are swapped;
Repeat the process as long as there are pairs of objects that need to be swapped;
The process terminates by performing an extra run when no further swaps are
needed;

*Also, accept less technical descriptions, the question does not ask for detail or
a low level algorithm.*

An example is shown below.

Run through array comparing side-by-side values;
Swap if needed;
Thus Bubbling the largest to the end;
Reduce array by 1;
Repeat;
Until only one value left;

[6 marks]

Total: [20 marks]

3. (a) If a plane's estimated arrival time is changed;
The FIDS display system could be updated automatically;
Award full marks [2 marks].

Accept answers such as the following for [1 mark].

Airport needs to know changes to inform passengers;

[2 marks]

- (b) *Possible answers.*

(i) *Award up to [1 mark max].*

Fibre optic / Wi-Fi/ WLAN/Ethernet(Cable);

[1 mark]

Do not accept LAN, WAN, etc.

(ii) *Award up to [2 marks max].*

(Fibre optic) would allow fast communication around the network;
Which is important for coordination between the different systems /
which is essential for critical systems;

(Wi-Fi) could allow airport / security personnel;

To access the network;

[2 marks]

- (c) Retina scans (any biometrics will do);
Will compare a person's characteristics with a central database;

[2 marks]

- (d) *Award up to [4 marks max].*

Modern display;

Give up-to-date information on plane movement;

Improved sound systems;

Provide clear audio information;

Wi-Fi systems;

Allow passengers to connect to the Internet;

[4 marks]

- (e) *In general, award marks as follows up to [5 marks max].*

Award [1 mark] for the name system;

Award [2 marks] for advantage and some discussion;

Award [2 marks] for disadvantage and some discussion;

For example, the baggage system:

Computerized systems move bags around the airport very quickly;

Allows passengers to make connections / avoids loss of bags;

However, if system breaks down there will be chaos;

As personnel to help will be limited;

Plus a substantiated overall opinion;

[5 marks]

continued ...

Question 3 continued

- (f) (i) Security would be able to see illegal access;
As tags will identify location of each passenger; **[2 marks]**

- (ii) Passengers who have not boarded;
Can be quickly located thereby avoiding delays; **[2 marks]**

- (g) (i) Would allow them to test different systems;
Without having to actually build them;
Saves considerable time (in the development cycle) / money;
As (supposedly) less chance of final system going wrong; **[4 marks]**

- (ii) Requirements change over time;
There may have been unforeseen problems with the original design;
Maintenance allows for beneficial changes to be made;
Maintenance keeps the system working efficiently; **[3 marks]**

- (h) Wi-Fi transmissions can be easily intercepted;
As they travel through the air / by anyone in the same area;
Encryption prevents the **understanding** of any intercepted messages;
*Don't accept answers related to customers being able to use the network
without paying.* **[3 marks]**

Total: [30 marks]
