Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

May 2016

Computer science

Higher level

Paper 2

25 pages

General marking instructions

1. Follow the markscheme provided, award only whole marks and mark only in **RED**.

2. Make sure that the question you are about to mark is highlighted in the mark panel on the right-hand side of the screen.

3. Where a mark is awarded, a tick/check (✓) **must** be placed in the text at the **precise point** where it becomes clear that the candidate deserves the mark. **One tick to be shown for each mark awarded**.

4. Sometimes, careful consideration is required to decide whether or not to award a mark. In these cases use RM™ Assessor annotations to support your decision. You are encouraged to write comments where it helps clarity, especially for re-marking purposes. Use a text box for these additional comments. It should be remembered that the script may be returned to the candidate.

5. Personal codes/notations are unacceptable.

6. Where an answer to a part question is worth no marks but the candidate has attempted the part question, enter a zero in the mark panel on the right-hand side of the screen. Where an answer to a part question is worth no marks because the candidate has not attempted the part question, enter an "NR" in the mark panel on the right-hand side of the screen.

7. Please ensure you check all scanned pages. The candidate may have answered more than one option. **If the candidate has attempted Option B**, please NR everything, record the script ID and email the details to the Principal Examiner and Subject Manager (emlyn.williams@ibo.org). Due to low candidate numbers, Option B will only be marked by the PE, and therefore no seeds will be produced for this Option.

8. **If a candidate has attempted more than one Option** within a paper mark all the candidate's work. RM™ Assessor will only award the marks for the higher scoring Option. Once all the work the candidate has attempted has been marked, please click "COMPLETE"; all the other questions from the other Options will auto complete to "NR" for "no response".

9. Ensure that you have viewed **every** page including any additional sheets. Please ensure that you stamp 'SEEN' on any page that contains no other annotation.

10. A mark should not be awarded where there is contradiction within an answer. Make a comment to this effect using a text box or the "CON" stamp.

Subject details: Computer science HL paper 2 markscheme

Mark allocation

Candidates are required to answer **all** questions in **one** Option. Total 65 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.
- 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**".

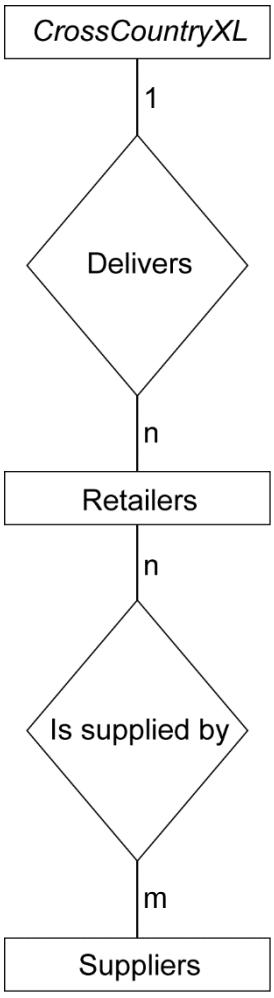
General guidance

Issue	Guidance
Answering more than the quantity of responses prescribed in the questions	• In the case of an "identify" question read all answers and mark positively up to the maximum marks. Disregard incorrect answers. • In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications. • In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation *etc*.

Option A — Databases

1.

(a)



[3]

*Award up to **[3 max]**.*
*Award **[1]** for correct 1:n relation.*
*Award **[1]** for correct n:m relation.*
*Award **[1]** for correct verbs (delivers, supplies). Accept equivalent verbs.*
*Award **[1]** for three tables (suppliers, retailers, CrossCountryXL).*
***Note:** Accept the Retailer:Supplier relation as 1:n.*

(b) Primary key must be unique;
Hence cannot be repeated; **[2]**

(c) (i) Oranges, bags, wallets, tomatoes; **[1]**
*Award **[1]** for all four items in any order.*

(ii) 34567890, 54959299, 45908919; **[1]**
*Award **[1]** for all three Supplier_ids in any order.*

***Note:** Accept any answer that lists the required data items.*

(d) *Award marks as follows up to **[4 max]**.*
*Award **[1]** for all relevant tables selected (supplier and supply).*
*Award **[1]** for all relevant fields selected.*
*Award **[1]** for correct condition.*
*Award **[1]** for correct link between tables.*

***Note:** SQL is not required, so accept algorithm style or list of clear points that describe the action required.*

From the tables SUPPLIER JOIN/AND/UNION SUPPLY
Return the field SUPPLIER.Name
Such that
SUPPLIER.Supplier_ID=SUPPLY.Supplier_ID
and SUPPLY.Nature_Of_Goods="P") **[4]**

(e) *Award **[1]** for stating a suitable measure and **[1]** for an explanation, for two measures up to **[4 max]**.*

Log-based recovery;
It is important that the logs are written prior to actual modification and stored on a stable storage media, which is failsafe;
This is done by keeping the log file on stable storage media or when a transaction enters the system and starts execution, it writes a log about it;

Back up files of the current database;
Refer to them at any time to reconstitute the database; **[4]**

2. (a) (i) The logical structure of the data (in the database); **[1]**

(ii) "50" is data;
"Attendance=50 days" is information; **[2]**
***Note:** Award **[2]** for any relevant examples that distinguish between data and information related to the question. **DO NOT** award marks for general statements that do not relate to the question.*

(b) (i) *Award **[1]** for an example and **[1]** for an explanation, up to **[2 max]**.*

The lost update problem;
This problem occurs (for example) when two operations by teachers, accessing the same items, have their operations interleaved in a way that makes the value of some items incorrect;

The temporary update problem;
This problem occurs when one transaction by a teacher updates a database item and then the transaction fails for some reason. Meanwhile the updated item is accessed by another transaction before it is changed back to its original value;

The incorrect summary problem;
If one transaction is calculating an aggregate of term grades/summary function on a number of database items while other transactions are updating these items, the aggregate function may calculate some values before they are updated and others after they are updated; **[2]**

***Note:** Accept answers that describe the above with an example even if the problem is not explicitly stated.*

(ii) Isolation;
Prevents the modification of the same data item by two different transactions;
Consistency;
Resolves the temporary update problem; **[4]**

(c) Rollback signals that the transaction has ended unsuccessfully;
For instance the calculation of the grades (or any form of calculation in the school management system);
So that any changes that the transaction may have applied to the database must be undone; **[3]**

(d) *Award up to **[2 max]***.
Changes applied to that database by a committed transaction must persist in the database (like student attendance or term grade changes);
These changes must not be lost because of any failure;
This is the responsibility of the recovery subsystem of the DBMS; **[2]**

3. (a) *Award up to **[2 max]***.
Granting access rights/password levels for some employees of the science research firm;
Removing access rights/password levels for some employees who (consistently) break rules;
Make the data read-only;
User accounts assigned appropriate security clearance level depending on roles; **[2]**

(b) One single breach of security in a large sized database could cause the firm's research/clients to be leaked to a competitor; **[1]**
Note: *Candidate must state consequence for the mark to be awarded.*

(c) *Award **[1]** for suggesting a method and **[1]** for relating it to EMP_DEPT, for two methods up to **[4 max]**.*
In table EMP_DEPT remove EmpName or personal data from reports prepared for public distribution;

In EMP_DEPT anonymize sensible individual data, such as masking some SSN characters;
When the company has to release reports that involve their employees;
Note: *Do not accept encryption or restricted access – these do not apply to output.* **[4]**

(d) *Award up to **[4 max]***.
Mining: Use of patterns to identify trends;
Matching: Comparing two sets of collected data;
Mining: To extract buried or previously unknown pieces of information from large databases;
Matching: To find errors in data;
It allows those holding large amounts of data to perform precise searches; **[4]**

(e) (i) *Award up to **[2 max]**.*
Minimizes redundancy;
Minimizes insertion/deletion;
Remove transitive dependencies;
Among non-primary key attributes; **[2]**

(ii) *Award up to **[3 max]**.*

*Award **[1]** for indicating keys in each of the tables returned.*
*Award **[1]** for at least two tables are correctly in 3NF.*
*Award a further **[1]** for all tables correctly in 3NF.*

Example 1

EMP1 : [EmpName, SSN*,DOB]
EMP2 : [EmpName*,Address,Salary,DepName]
DEP : [DepName*,DepNo,DM-SSN]

Example 2

EMP1 : [EmpName, SSN*,DOB, Address, Salary]
EMP2 : [EmpName*,DepName]
DEP : [DepName*,DepNo,DM-SSN] **[3]**

4. (a) *Award **[1]** for any valid classifications, up to **[3 max]**.*
Customers who prefer a particular type of fast food;
The day of the week on which the maximum sales occur;
Age groups and preferences relation;
The typical combinations that sell well;
Price ranges that sell more; **[3]**

(b) The timestamp specifies the time and date of an operation;
Data can be accessed in the warehouse by any date and time order or grouping; **[2]**

(c) *Award up to **[3 max]**.*
Making identifiers unique;
Convert null values into standardized “Not Available/Not Provided” value;
Convert phone numbers/ZIP codes to a standardized form;
Validate address fields, convert them into proper naming, *eg* Street/St/St./Str; **[3]**

(d) *Award up to **[4 max]**.*
Data warehouses exist as persistent storage, while a view is on demand;
Data warehouses are not relational, while a view is relational;
Data warehouses can be indexed to optimize performance, while a view cannot be indexed independent of the database;
Data warehouses provide specific support of functionalities, while a view cannot;
Data warehouses provide large amounts of integrated and temporal data;
Views are extracts of the database; **[4]**

(e) *Award up to **[2 max]**.*
Disable any constraints and indexes before the load;
And enable them back only after the load completes;
The referential integrity needs to be maintained by ETL tools to ensure consistency as a lack of it can create incomplete data that might result in erroneous results after the loading process; **[2]**

(f) (i) *Award up to **[2 max]**.*
Polymorphism;
Inheritance;
Encapsulation; **[2]**

(ii) *Award **[1]** for each advantage/disadvantage identified, up to **[2 max]**.*
*Award **[1]** for each advantage/disadvantage explained, up to **[2 max]**.*

Advantages:

Less Maintenance;
If processes within the system are encapsulated, their "behaviours" can be reused and incorporated into new ones;

More code reusability;
When a new object is created, it will automatically inherit the data attributes and characteristics of the class from which it was taken. The new object will inherit the data and behaviours from all superclasses in which it participates;

Larger variety of data types;
Unlike traditional databases (such as hierarchical, network or relational), object-oriented databases capable of storing different types of data, for example, text, numbers, pictures, voice video etc;

No impedance mismatch;
A single language interface between the Data Manipulation Language (DML) and the programming language overcomes the impedance mismatch. Most OODBMSs provide DML that is computationally sufficient and adequate compared to SQL, which is the standard language of RDBMSs;

Disadvantages:

No support on views;
OODBMSs do not provide a "view" mechanism, which provides many advantages such as data independence, security, reduced complexity, and customization;

No support for security;
OODBMSs are not adequate on security issues;

Users cannot grant access rights on individual objects or classes;

No universal data model;

No universally agreed upon data model for an OODBMS;
This disadvantage is seen as a major drawback, and is comparable to pre-relational systems. **[4]**

Option B — Modelling and simulation

5. (a) *Award **[1]** for a suitable way of storing **each** of the four variables.*
For example:
Size as an integer/number *eg* "100" for "100 m^2";
The type or property – need two identifiers *eg* 1 for house and 2 for apartment (accept letters);
The district – need different identifiers *eg* 1 for district 1, *etc* … (accept letters);
The condition of the property – group into categories *eg* 1 = "very poor", 2 = " poor" *etc* (accept letters); **[4]**

(b) *Figures will depend on the groupings made in part (a).*
For example:

	A	B	C	D	E	F
1	Size	District	Type	Condition		Estimated Price
2	100	4	1	3		=100*2000*0.9*1
3						

*Award **[1]** for correctly setting out the variables and estimated price.*
*Award **[1]** for explaining the formula for **each** of the selections in B,C and D.*
For example:
If B = 1 fb = 1.25 *etc*
If C = 1 p = 2000 else p = 2300
If D = 1 cp = 0.9 *etc*
*Award **[2]** for combining these with A to give correct formula.*
F = A*fb*p*cp **[6]**
***Note:** Accept using lookup tables for "type", "district" and/or "condition", provided it is explained. Award **[1 max]** for drawing a table with ALL components but not indicating how calculations are performed.*

(c) *Award **[2]** for comparing the price sold and the estimated price over a large number of sold properties and **[1]** for considering date.*
For example:
For each property sold, compare estimated price with selling price;
If there is consistently a difference over many sales then rules need to be changed;
If many properties take too long to sell, rules could need to be changed; **[3]**

(d) *Award **[1]** for a suggestion and **[1]** for an expansion, for two suggestions up to **[4 max]**.*

For example:
Compare the characteristics of those that have been sold with those that have been on the market for a long time;
This can be done by grouping together houses/apartments and seeing if the proportion of sold and not sold is different;
Compare the length of time before it is sold and for each characteristic (and/or combination);
To check if those in poor condition (for example) take longer to sell; **[4]**
***Note:** Accept references to comparisons, over a time/date, which are stated in (c).*

6. (a) *Award **[1]** for defining a model and **[1]** for relating it to the scenario.*
For example:
A model is a set of data and mathematical rules that define a specific situation;
In this case it is the set of options available at each station at different times of the day and the way in which they are used;

*Award **[1]** for defining a simulation and **[1]** for relating it to the scenario.*
For example:
A simulation uses a mathematical model to see the effect of different inputs through the system under different conditions and the effect of changing the model;
In this case the number of machines could be increased and the effect seen, as data for each day is put in the simulation; **[4]**

(b) (i) *Award up to **[2 max]**.*
The number of potential passengers;
On each day in the week / at each time of day;
The time taken to perform operations at machines and ticket offices;
Cost of machines;
Cost of hiring manual operators;
Cost per unit time/capital cost/running costs *etc*; **[2]**

(ii) *Award up to **[2 max]**.*
This can be collected from historical data;
From machines and ticket offices giving date/time bought and date/time of journey;
Central database which has all journeys recorded;
Observation over many different days, and times of day, could be used; **[2]**
***Note:** Accept an answer of "questionnaires" only if sensibly contextualized.*

(c) *Award up to **[2 max]**.*
Using the passenger data from above;
Change the number of machines;
Change the number of ticket offices;
Observe the flow of passengers;
At each time/day period; **[2]**

(d) *Award **[1]** for a criterion and **[1]** for an expansion, up to **[2 max]**.*
Time taken for each of the operations by passengers;
During rush hour and quiet times;
Cost of installing new machines;
Compared to manning the ticket offices for more or fewer hours; **[2]**

(e) *Award **[2]** for discussing a benefit/similarity between the stations and **[2]** for discussing why the results are not transferrable.*
For example:
The structure of the simulation could be the same;
Although the data throughput would be different depending on the situation and size of the station;
The results are likely to be very different;
The data would need to be collected from each station separately; **[4]**

7. (a) *Award **[2]** for an advantage that is expanded and **[2]** for a disadvantage that is expanded, up to **[4 max]**.*

Advantages:
Many different situations can be presented as opposed to the real life situation where experiences are limited to what is presented;
More time can be spent with the simulator than flying hours with a pilot;
It is less dangerous as the situations are not real and wrong reactions do not have real consequences;

Disadvantages:
Virtual training does not give the same sense of reality and quick reactions may not be the same in real situations;
The training pilot would benefit from the first-hand knowledge of the experienced pilot;
There is a cost in installing and maintaining the simulator; **[4]**

(b) *Award **[2]** for an outline that includes both software and hardware.*
For example:
Fast processor/dual processor with separate graphics capabilities;
Complex image processing software; **[2]**
***Note:** Accept hardware that refers to actuators or moving cabins.*

(c) *Award up to **[3 max]**.*
Landscape is not flat and positions must be shown in 3D;
The pilot will need to react quickly in three dimensions/there is a vertical component;
And the landscape will also have to respond to the actions/in real-time;
And give a new 3D image each time; **[3]**

(d) *Award up to **[3 max]**.*
The image is held in mathematical form;
In many parts which can be changed/manipulated as needed;
Visualization involves translating the mathematical image into a human understandable form;
Rendering as needed;
Using many techniques, such as ray tracing, to give depth/light and shade; **[3]**

8. (a) *Award up to **[5 max]***.
A set of possible paths from point to point is produced;
Measured against a fitness function;
The least suitable are discarded;
Replaced by a set created from the better paths;
For example mutation/cross over (accept other suitable methods);
This is repeated until the best solution found; **[5]**

(b) *Award up to **[6 max]***.
In supervised learning, the network is provided with inputs;
And provides outputs which can be tested against the known output;
Until a system is found which gives the correct answer;
In unsupervised learning the answer is not known in advance;
But evolved via pattern recognition;
In this case the learning is unsupervised;
As the network is finding an unknown pattern; **[6]**

(c) Rules/syntax;
Vocabulary;
Repetition; **[3]**
Note: *Accept other correct answers provided they are regulated by rules.*

(d) *Award marks as follows, up to **[6 max]***.
*Award **[2]** for a description of the way humans learn language*.
*Award **[2]** for a description of machine learning*.
*Award **[2]** for a comparison/conclusion of differences*.

For example:
Human cognitive learning/complex thought process;
Include past experience of the ways in which words are used / ambient senses such as facial expressions when words are used *etc*;

Machine learning involves following preset rules;
Combined with heuristics – building up experience over time;
Using probabilities to place words/phrases correctly;

These different approaches to learning make it difficult for machines to use/interpret language in the same way as humans;
Research into cognitive learning/human thinking could mean that (given time), machines could be taught natural language; **[6]**

Option C — Web science

9. (a) (i) HTTP;
SSL/TLS; **[2]**

(ii) *Award up to **[4 max]**.*
HTTP permits the transfer/exchange of data *etc*;
Over the internet/network / Between client and server;

SSL/TLS allows for this data to be encrypted/makes use of PKI;
By allowing the (secure) exchange of (session) keys;
Authentication of server (and client); **[4]**

(b) When a user requests a specific URL / When a link is activated;
The web browser passes this data (website domain name) to a domain name server;
Which looks up the corresponding IP address (or passes the request upwards);
This IP address is then returned to the browser by the server; **[4]**

(c) (i) *Award up to **[4 max]**.*
The file is sent to the PHP interpreter to be processed (because it is recognized to be a PHP script);
The database, "`library.db`", is accessed;
The data `$POST["author"]` is extracted from the form;
The database query is carried out (using the above data);
The results (as part of an HTML page) are sent back to the client (user); **[4]**

(ii) *Award up to **[4 max]**.*
Prevents direct access to the server's database;
Therefore protecting potentially valuable information;

Allows access rights/restricted access to data;
Permitting data to be seen by only those who are allowed to do so;

Hides the scripts from the client;
Thereby preventing malicious altering of the code; **[4]**

10. (a) Techniques used with the aim of improving the ranking of a web page;
Using methods that are considered misleading/unethical/designed to gain an unwarranted/unfair advantage; **[2]**

(b) The technique is keyword stuffing/loading a web page with often irrelevant keywords;

*Award a further **[2]** for each valid consequence, up to **[4 max]**.*
The ranking might improve (initially) for searches;
As there are many references to the search term;

Users will end up avoiding the site;
As it may provide a poor user experience;

Search engines may reduce its ranking/not rank it at all;
As the technique is considered malpractice/black-hat; **[5]**

(c) *Award up to **[3 max]**.*
Pages which have high quality content (authorities) will (almost certainly) have a high ranking themselves;
Search engines will increase the ranking of a page if the in-links have a high ranking / Ranking algorithms include the ranking of in-links in their calculations;
Example: A scientific company's web page pointed at by the *Scientific American*;

Alternative answer:
Quality sites refer to sites that are recognized authorities (*eg New York Times*, *BBC*, *Google etc*);
Or specifically related to the content of the site (*eg* home improvement site linking to a supplier's website);
These links will receive a higher factor/be considered more favourable when calculating rankings;
As they are considered more relevant (than unrelated sites); **[3]**

11. (a) (i) *Award **[1]** for any answer that gives **EITHER** the idea of being able to use computers (computing power) wherever you are **OR** that computing would be available on countless different devices.* **[1]**

(ii) *Award up to **[2 max]**.*
The number of bits in an IP address determines the number of uniquely addressable devices;
Each version of IP fixes the number of bits available (which can be different in different versions);
So the current version used must provide sufficient bits if the IoT is to happen;

Alternatively:
Browsers *etc* need to (complete the) switch from IPv4 to IPv6;
As IPv4 will not have enough available addresses to allow the IoT to happen; **[2]**

(iii) *Award **[1]** for describing an example that is connected with the scenario.*
*Award **[1]** for an adequate description of the device(s) being used.*
*Award up to **[2 max]** for a credible explanation of how its use helps the company.*

Example:
The rental company can keep track of the condition of its cars;
Microprocessors will be monitoring the state of the engines;
These microprocessors will send this information to the company's offices;
As each microprocessor is identifiable, so is each car; **[4]**

(b) *Award up to **[2 max]** for a good description of each limitation, for **two** limitations up to **[4 max]**.*
The result returned may not be accurate/may be a false-positive;
As quotes/common phrases may be included;

It may falsely return a negative result (not plagiarized);
As the plagiarized material is not in its database; **[4]**

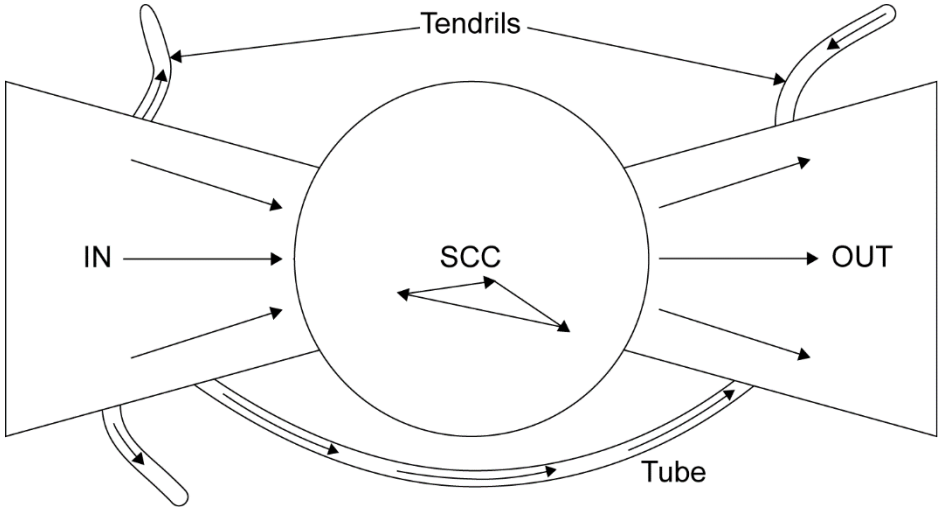
(c) *Award up to **[2 max]** for a good explanation of the democratic web/net neutrality.*
*Award up to **[2 max]** for any valid argument in favour (award **[1]** for a valid attempt);*
*Award up to **[2 max]** for any valid argument against (award **[1]** for a valid attempt);*

Net neutrality/democratic web refers to the equal treatment;
Of providers/users;

For:
This means that (for example) small enterprises have the same opportunity to promote their services as major international companies;
Which allows fair competition for all;

Against:
Normal business practices provide different levels of service for different costs;
A precedent has already been set by users paying for different levels of service *(for example)* download speeds, monthly data download limits; **[6]**

12. (a)



*Award **[1]** for a diagram with the correct general shape (see above).*
*Award **[1]** for correctly labelling each of the following up to **[3 max]**.*
- IN
- OUT
- SCC
- Tube
- Tendril **[4]**

(b) (i) Each web page would be a node/vertex on the graph;
Links between pages would be shown by the edges/lines;
Arrows would indicate the direction of the links (1–way / 2–way); **[3]**

(ii) ***EITHER***
As the PageRank of a page is calculated, this ranking will affect/change pages to which it links;
This (chain-reaction) will inevitably affect the ranking of pages linking to the original page;
Thereby requiring a new calculation;

OR
The PageRank of a page is a calculation that is dependent upon the number of inlinks/PageRank of inlinks;
As both of these are continually changing;
The calculation must be continually repeated; **[3]**

13. (a) *Award **[1]** for an answer that clearly shows what is meant by the semantic web.*
*Award **[1]** for an answer that clearly shows what is meant by an ontology.*
*Award up to **[2 max]** for an answer that shows the relationship between the two.*

A semantic web is one in which all data can be logically searched by algorithms/ "understood" by computer programs;
An ontology (in this context) is a prescribed way of describing data;
For the aims of the semantic web to be met, it is essential that consistent ways of describing data are agreed upon so that search algorithms will return a complete valid, set of results; **[4]**

(b) *Award **[1]** for correctly defining ambient intelligence.*
*Award **[1]** for correctly defining collective intelligence.*
*Award **[1]** up to **[4 max]** for putting the following terms in context:*

Nanotechnology
Sensors
Smart phones
Centralized servers

*Award **[1]** for examples that put any of the above in a suitable context and **[1]** for outlining the benefits to the patient in that example.* **[6]**

Option D — Object-oriented programming

Note: *Perfect syntax is not required answering algorithm questions.*

14. (a) Association/dependency as `Client` has a `Room`; **[1]**

Note: *Accept aggregation, as defined in the subject guide.*
Accept "has a `Room` *object" / "contains a* `Room` *object"/ uses a* `Room` *object (and other similar wording).*

(b) *Award **[1]** for a three-tier box, **[1]** for correct class name, **[1]** for all variables correct and **[1]** for all get/set methods correct.*

class: Room
int: roomNumber int: beds double: price Boolean: empty
setRoomNumber(int n) setBeds(int b) setPrice(double p) setEmpty(Boolean e) getRoomNumber() getBeds() getPrice() getEmpty()

[4]

Note: *Accept* `int` *for price.*

(c) *Award **[1]** for identifying an advantage and **[1]** for an elaboration.*
Advantage:
Encapsulation in Dates;
(which implies) abstraction of Dates;
Explanation:
(encapsulation) combines data and methods;
So as to reuse the code for multiple dates. **[2]**

(d) *Award marks as follows up to **[6 max]**.*
*Award **[1]** for correct method declaration including data type*
Award [1] correct return.
*Award **[1]** for creating `available` array (or other method to record `empty`).*
*Award **[1]** for correct loop through `allRooms[]`.*
*Award **[1]** for checking for two beds.*
*Award **[1]** for checking room is empty.*
*Award **[1]** for correctly recording available room.*

For example:

```
public int[] findRooms()
{
  int[] available;
  available = new int[100];
  int j;
  j=0;
  for(int i=0; i<100; i++)
  {
    if(allRooms[i].beds == 2 && allRooms[i].empty)
    {
      available[j] = allRooms[i].roomNumber;
      j = j + 1;
    }
  }
  return available;
}
```
[6]

Note: *Accept a String being returned. It is acceptable for the student to not use an array but to concatenate a String.* Accept use of ArrayList. Accept void/ouput method.

(e) *Award marks as follows up to **[8 max]**.*
*Award **[1]** for correct method declaration /*
`void` with `output` or `double` with `return`.
*Award **[1]** for using both dates.*
*Award **[1]** for calling `stayDays()`.*
*Award **[1]** for correct values entered into `stayDays`.*
*Award **[1]** for getting correct price.*
*Award **[1]** for correct room accessed.*
*Award **[1]** for output of correct cost.*
*Award **[1]** for output of bill including all details.*

Note: *Be lenient on the names given to method eg `getPrice()` as compared to `price` etc. Accept methods that return a double.*

```
public void bill()
{
  int stay;
  stay = Dates.StayDays(arrive, leave);
  double cost = getBedroom().getPrice()*stay;
  System.out.println("Client" + name + "room" +
                              getBedroom().getRoomNumber());
  System.out.println("date arrived" + arrive +
           "date booked out" + leave + "cost = " + cost);
}
```
[8]

15. (a) *Award marks as follows up to **[7 max]**.*
*Award **[1]** for use of `extends`.*
*Award **[1]** for `groupName` declared as `private`.*
*Award **[1]** for ALL variables correct (including `String groupName`).*
*Award **[1]** for use of `super` with correct parameters.*
*Award **[1]** for setting `groupName = g`.*
*Award **[1]** for correct set `groupName`.*
*Award **[1]** for correct get `groupName`.*

```
public class GClient extends Client
{
  private String groupName;
  public GClient(int id, String c, Dates dateIn, Dates dateOut,
                   Room r, String g)
  {
    super (id, c, dateIn, dateOut, r);
    groupName = g;
  }
  public void setGroupName(String groupName)
  {
    this.groupName = groupName;
  }
  public String getGroupName()
  return groupName;
}
```
[7]

(b) *Award marks as follows, up to **[6 max]**.*
*Award **[1]** for a correct instantiation of `Group`.*
*Award **[1]** for searching for empty rooms.*
*Award **[1]** for looping through all group members.*
*Award **[1]** for allocating `roomNumber` and storing it in `allRooms[]`.*
*Award **[1]** for setting `empty` in `Room` object to `false`.*
*Award **[1]** for correct instantiation of `GClient` object with information including room number and group name.*

For example:
`Group` object instantiated `withname = Happy Travellers` and `number = 15`;
Use `findRooms()` to search for empty rooms with 2 beds;
For each of 15 in the group…
find next room in list of available;
status of room changed from empty;
room number passed to `gRooms`;
instantiate new `GClient` with name of group and room number; **[6]**

(c) *Award marks as follows, up to **[5 max]**.*
*Award **[1]** for correct declaration of all variables (not method signature).*
Accept `int` for `roomCost` and `totalCost`.
*Award **[1]** for setting `totalCost` to zero.*
*Award **[1]** for looping through `gRooms`.*
*Award **[1]** for correct `roomNumber` found in `allRooms[]`.*
*Award **[1]** for `getPrice` from `Room`.*
*Award **[1]** for adding to `totalCost`.*
*Award **[1]** for outputting/returning `totalCost`.*

Note: *Be lenient on method names used.*
Accept methods that return a double.

Example:

```
public void bill (int[] gRooms)
// method to calculate bill for the group
{
  double totalCost = 0;
  double roomCost;
  for (int i=0;i<number;i=i+1)
  {
    for (int j=0;j<100;j=j+1;)
    {
      if (gRooms[i] == allRooms[j].getRoomNumber())
      {
        roomCost = allRooms[j].getPrice();
      }
    }
    totalCost=totalCost + roomCost;
  }
  System.out.println(totalCost);
}
```
[5]

16. (a) *Award up to **[2 max]**.*
Thorough testing;
Cite sources;
Update documentation; **[2]**

(b) *Award **[1]** for identifying a feature and **[1]** for an expansion, for two features up to **[4 max]**.*
Unicode;
International character sets;
Portable (in Java);
Changes are straightforward (output messages, currencies *etc*); **[4]**

17. (a) *Award marks as follows, up to **[6 max]**.*
*Award **[1]** for setting `aClient` at head, if list is empty.*
*Award **[1]** for setting `Boolean` flag/break or similar.*
*Award **[1]** for checking which date is greater.*
*Award **[1]** for checking for equal dates.*
*Award **[1]** for insertion in correct position.*
*Award **[1]** for continuing through `Bookings` until found.*
*Award **[1]** for adding, if no insertion point found.*

Note: *Be lenient on the naming of methods and objects, and it is not required that students use index when adding to the list.*

```
public void newClient (Client aClient, LinkedList Bookings)
{
  boolean found = false;
  int index = 0;
  if (Bookings.isEmpty())
  {
    Bookings.addFirst(aClient);
  }
  else
  {
    Client nextClient = Bookings.getFirst();
    while ((nextClient!=null) && (!found))
    {
      if (Dates.equalDate
           (Dates.compareDate(nextClient.arrive,aClient.arrive),
             aClient.arrive))  ||
         (Dates.equalDate(nextClient.arrive, aClient.arrive))
      {
        Bookings.add(index, aClient);
        found = true;
      }
      else
      {
        index = index + 1;
        nextClient = Bookings.get(index);
      }
    }

    if(!found)
    {
      Bookings.addLast(aClient);
    }
  }
}
```

[6]

(b) *Award marks as follows, up to **[6 max]**.*
*Award **[1]** for creating a new array of some reasonable dimension.*
*Award **[1]** for looping through `Bookings` until date is past today;*
*Award **[1]** for putting `Client` in array and updating next position.*
*Award **[1]** for removing from `Bookings`.*
*Award **[1]** for getting next correct Client from `Bookings`.*

Sort may be bubble after the array is filled or by placing into the correct order when placed. In either case:

*Award **[1]** for double loop.*
*Award **[1]** for correct comparison.*
*Award **[1]** for a correctly sorted array in order of `dateOut`.*

Example 1 – with bubble sort

```
void todayClients(Dates today)
{
  int i = 0;
  int index = 0;
  Client[] arrivalsToday = new Client[100];
  Client c = Bookings.getFirst();       // or similar
  while (c!=null) && (Dates.equalDate(c.dateIn, today))
                          //checks list not empty and date is today.
  {
    arrivalsToday[i] = c;          //add to array
    i=i+1;
    Bookings.remove(index);        //remove from list
    c = Bookings.get(index);       //get next in list
  }
  for (int j=0; j<i-1; j=j+1)
  {
    for (int k=0; k<i-1-j; k=k+1)  //double loop for bubble sort
    {
      if (Dates.equalDate(Dates.compare(arrivalsToday[k].dateOut,
        arrivalsToday[k+1]).dateOut, arrivalsToday[k+1]))
                              //compare adjacent dateOut
      {
        temp = arrivalsToday[k];
        arrivalsToday[k] = arrivalsToday[k+1];
        arrivalsToday[k+1] = temp;     //swap
      }
    }
  }
}
```

Example 2 – Placing in order

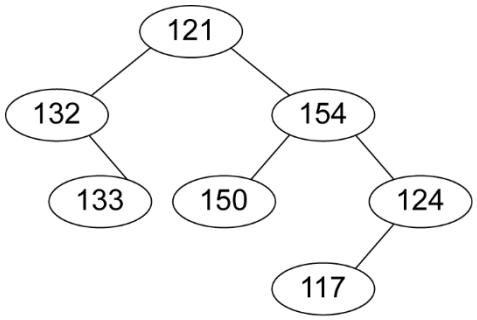
```
Client[] todayClients(Dates today)
{
  int count = 0;               // number of bookings added to array
  int i;                       // insertion point
  boolean found;
  Client[] arrivalsToday = new Client[100];

  Client c = Bookings.getFirst();           // or similar
  while ((c!=null) && (Dates.equalDate(c.dateIn, today)))
  {
    Bookings.remove(0);
    i = 0;
    found = false;
    while ((i<count) && (!found))       // find insertion point
    {
      if (Dates.equalDate(Dates.compareDate(c.dateOut,arrivalsToday[j]),
                          c.dateOut))
      {found = true; }
      else { i=i+1; }
    }

    for (int j=count; j>i; j=j-1)       // correct shifting
    {
      arrivalsToday[j] = arrivalsToday[j-1];
    }

    arrivalsToday[i] = c;               // insert c
    count = count + 1;                  // increment count
    Client c = Bookings.getFirst();     // get the next one
  }
  return arrivalsToday;                     // not required
}
```

Note: *For an empty array the two inner loops will not execute.*

[6]

(c) *Award up to **[4 max]***.
Can use the `arraylist` library class;
Do not have to declare size of the array;
Because the `arraylist` automatically resizes itself;
Do not need to identify the index;
Because library class auto increments for you;
Could have used “sort list” instead of finding the correct place to insert object; **[4]**

(d)



[4]

*Award **[1]** for correct diagram.*
*Award **[2]** for correct steps for a node;*
*Award **[1]** for using data in diagram to relate to recursion;*

Start at root (121)
inorder:
Go left as far as possible
if no left take node (132)
go right (133)
if no left or right go back up tree passing all taken
back to inorder

Hence "recursion" as each node calls the same inorder method.

***Note:** Accept any explanation related to the diagram which shows how the nodes are visited and recursively checked for leaf nodes giving the correct order.*