GCE MARKING SCHEME

COMPUTING
AS/Advanced

SUMMER 2012

INTRODUCTION

The marking schemes which follow were those used by WJEC for the Summer 2012 examination in GCE COMPUTING. They were finalised after detailed discussion at examiners' conferences by all the examiners involved in the assessment. The conferences were held shortly after the papers were taken so that reference could be made to the full range of candidates' responses, with photocopied scripts forming the basis of discussion. The aim of the conferences was to ensure that the marking schemes were interpreted and applied in the same way by all examiners.

It is hoped that this information will be of assistance to centres but it is recognised at the same time that, without the benefit of participation in the examiners' conferences, teachers may have different views on certain matters of detail or interpretation.

WJEC regrets that it cannot enter into any discussion or correspondence about these marking schemes.

CG3

01 Any one of: 1

- a printer queue
- a keyboard buffer
- a download buffer
- a processor scheduling queue

[**Note:** other computer applications are possible so **not** e.g. a queue of patients at doctor's, though could be a computerised version of this]

Why: In each case, because the natural / desirable processing order is first in first out (or e.g. job waiting longest should be printed next) 1
But **just** putting e.g. "because it's first in first out" is not enough

---

02 Any 1 of:

- subprogram return addresses 1
- recursion values
- short-term arithmetical results
- reversing a queue / list
- undo function
- reverse Polish calculations - **accepted not expected**

[**Note:** other computer applications are possible, but **not** e.g. a pile of plates]

Why: In each case, because the natural / desirable processing order is first in last out (or e.g. last address placed on stack needs to be first one to be accessed, etc)
But **just** putting e.g. "because it's first in last out" is not enough 1

---

03 Any three from:

- Cannot see / feel / try-on the item (for a tangible item) / may not be as advertised
- Lacks the social aspect of shopping
- Need access to a debit card / credit card
- Can't get the item immediately (tangible item)
- There is often a delivery charge (tangible item)
- If not at home when delivered you may need to visit post office, etc (tangible item) / you need to be at home when delivered
- Something may get lost or broken in the post (tangible item)
- May need to return item - more difficult than going back to shop (tangible item)
- Website may be fraudulent - may not receive goods (Separate points
- Website may be fraudulent - credit card details may be stolen if well argued)
- May have an impact on traditional shops closing down. 3

Not health related risks e.g. RSI / lack of exercise

---

04 **Half-duplex** transmission is when data can be sent in either direction (along a computer network / pathway / cable), but only in one direction at a time. 1

Full-duplex transmission is when data can be sent in either direction (along a computer network / pathway / cable), in both directions at the same time / simultaneously. 1

---

05 A network protocol is necessary to enable devices/computers to communicate with each other, e.g. 1

- linking a printer to a computer
- linking mobile phones by Bluetooth
- using ftp/http/smtp/voip/pop3 to transfer data between devices 1

06 Data collision occurs when two sets of data are detected on the network simultaneously 1

Once detected, a computer waits for a short/random time then sends again 1

---

07 Parallel <u>transmission</u>: occurs when a number of bits are sent simultaneously / sent over multiple paths / lines. 1

Parallel <u>processing</u>: the simultaneous use of several processors/cores to perform a single task 1

PP could be useful where an extremely large/complicated calculation is being carried out (e.g. weather forecasting, image processing…) 1

---

08 Any 5 from:

- GUI system is usually easier to learn for a novice user
- GUI system is usually more intuitive to use once learned e.g. icons relevant to the application
- may be similar to other packages with which users are familiar
- there is no need to remember complex text commands
- it is easier to cut and paste between applications
- users can customise desk-top, e.g. creating shortcuts, etc
- GUIs often have a good help system / tutorials / wizards, etc 5

An example of an extended answer worth five marks is:

A GUI system is usually easier to learn for a novice user and is also usually more intuitive to use once learned. It may be similar to other packages with which users are familiar, which will also help to make it easy to learn. It may use icons, etc which are relevant to the application which make it more transparent to the user. There is no need to remember complex text commands. It is usually easy to cut and paste between applications. In many cases, the user can customise the desk-top, etc, and many GUIs have a good help system and may come with a tutorial.

[**Marking**: Note: question asks about **<u>benefits</u>**. No marks for features or drawbacks of a GUI]

09

- person's voice (print) originally captured, digitised and stored
- on attempted entry, person's voiceprint captured (and digitised)
- digitised data compared (and entry allowed if successful) 3

An example of an extended answer worth three marks is:

Voiceprint recognition is the process of capturing the person's voiceprints then digitising and storing this data on a computer system. When entry is attempted, the voiceprint of that person is captured. The two data items are compared, with entry being allowed if there is a match.

---

10 XOR Truth Table (either format)

Worked example must show **data** and **key** correctly combined using **XOR** then combined using **XOR** to retrieve original data. 1

Eg:

D(ata)	10111101	
K(ey)	10001111	
D **XOR** K	01111010	= **Y**
Y **XOR** Key	10111101	2

Marking 1 for correct XOR on Data and Key
1 for correct XOR on the result with Key again

---

11 An algorithm is a (finite) <u>set of rules / instructions</u> 1

to solve a (specific) <u>problem</u> 1

---

12 A recursive algorithm is one which calls itself. 1

It must also have a "base case" (or "simple case") to allow it to terminate / terminating condition. (**Condone** idea of unwinding) 1

<u>Quicksort: why often used:</u>
Because it is the fastest / most efficient type of sort algorithm 1

<u>Quicksort description</u>:

- An item/pivot selected (which item is unimportant) **<u>Condone</u>** "middle" 1
- Produce two new lists of smaller and larger numbers 1
- Repeat above points on new sub lists (recursively) until sorted 1

Accept examples/diagram instead

---

13 [In this order]

- The O/S suspends current interrupt routine 1
- It runs the new higher priority interrupt routine 1
- The O/S returns to original interrupt routine and continues 1

Description of high priority interrupt involving any of:

- Impending data loss 1
- Impending hardware / software failure
- Detection of imminent power failure
- **<u>Allow</u>** run time error

---

14 Random Access file is a computer file where: 6x1

- Physical location for new record is calculated from the key field / user-Id
- A hashing algorithm is used for this calculation to find the location
- If data collision /something there, the record is stored instead in an overflow area
- Data in the overflow area is normally stored and searched in a linear manner
- File may need reorganising (and new hashing algoriyhm) if overflow becomes too large
- Existing records are accessed in the same way.

A well explained example could gain all six marks.
An example of an extended answer worth six marks is:

A random access file is a computer file where the physical location for a new record is calculated from the data in the record's key field, the user id in this example. A hashing algorithm is used for this calculation. If this location is empty, the encrypted password is stored there, but if the location is already occupied with data, the password is normally stored instead in an overflow area. Data in the overflow area is normally stored and searched in a linear manner. When access is required to an existing record, the process is similar to above – the same hashing algorithm is used to derive the location, with the overflow area being accessed if necessary. If the overflow area becomes too large access times may lengthen and the file may need re-organisation.

---

15 -9 (CAO) 1

---

16 One method is:

- From RHS, rewrite it up to and including the first one 1
- Change other 1 digits to 0 and 0 digits to 1 1
- Correct working and answer for example
  `00001000 -> xxxx1000 -> 11111000` 1

Alternatively

- Flip the bits 1
- Add one 1
- (Ignore carry (ninth bit))

(Other methods equally acceptable)

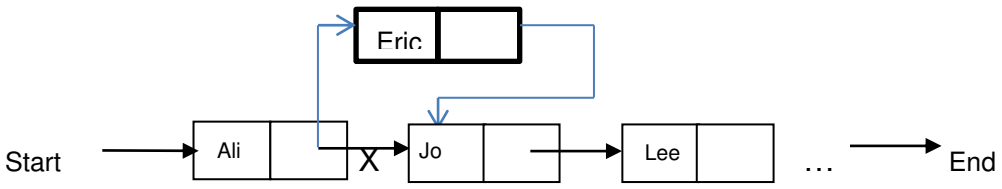
---

17 Any 1 of:

- enables computer (systems) to communicate with each other easily 1
- use of (mainly) just one code avoids confusion

"L" = 76 1

18



Marking:
Crossing out / non copying of link Anu -> Jo 1
Insertion of name in correct place with arrow to it 1
Correct arrow to Jo 1

---

19 A robot is a computer controlled (**accept** programmed/programmable, etc) device(s) 1

which can carry out (a variety of) complex (useful / repetitive) tasks
(**accept** mimic human activity, etc) 1

In a car factory, a robot could be used for:
Welding, painting, moving of raw materials, finished components / vehicles, etc 1

Benefits
- Perform the required task more consistently than human worker
- Perform the required task more accurately / with finer tolerances than human worker
- Work 24 hours per day, 7 days a week
- Don't get tired / ill / go on strike / need tea break / toilet break
- Cheaper to run (no salary costs)
- Can work more quickly than human worker
- Factory may be able to run without heating / lighting
- Human workers will not be subject to dangerous fumes / heat, etc
- Less chance of employees developing computer related ailments such as RSI

4x1

Drawbacks
- Staff remaining will have to be highly skilled / need significant training, etc
- Lacks touch / sensitivity / versatility / intelligence of human worker
- Robots may not notice obvious/repeated errors in process / cannot adapt to changes
- Expensive to buy / set-up / install
- May require factory to be rebuilt / extended, etc
- May be expensive to maintain / repair
- Difficult to reset / change for new production run / take time to reset, etc
- Robots may not notice obvious situations such as factory on fire if no humans around
  Not "may break down" alone

[Marking of Benefits/Drawbacks: 1 mark for each point up to max of 4, but max of three if only Benefits or only Drawbacks attempted]

An example of an extended answer worth seven marks is:

A robot is a computer controlled mechanical device which can carry out a variety of complex useful tasks.

In a car factory, a robot could, for instance, be used for: welding, then painting car components and the moving of raw materials or finished vehicles around the factory.

There would be a large number of benefits to the owners of the car factory. For instance, the robot would:
- perform the required task more consistently and with finer tolerances than a human
- work 24 hours per day, 7 days a week
- be cheaper to run, since the system itself would have no salary costs
- work more quickly than human workers
- avoid workers being subject to dangerous fumes, etc

Drawbacks would include:
- there would be significant training cost for those staff who remain
- the robot may lack the versatility and intelligence of human workers
- the robots are expensive to buy
- the robots may be difficult to reset or change for new production run

20 A multiprogramming computer system is one where more than one job is held in the computer's main memory at the same time and can be processed in the computer's central processing unit (CPU) at (apparently) the same time. Multiprogramming is used to ensure the most efficient use of the CPU and prevent the CPU being idle while waiting for a slower peripheral. The real-time clock causes regular interrupts to create time-slices, which the operating system allocates to the various jobs: this process is called scheduling and is controlled by a scheduler program. Each job is checked sequentially to ensure that it gets its appropriate share of time – this is known as polling. To allow more than one job to be resident in the main memory at any one time, the memory needs to be separated into separate parts - this is called partitioning.

6

[When answers are given in well-expressed point form:
6 marks may be gained for six or more of the individual points listed below
5 marks may be gained for five of the individual points listed below
4 marks may be gained for four of the individual points listed below
3 marks may be gained for three of the individual points listed below
2 marks may be gained for two of the individual points listed below
1 mark may be gained for one of the individual points listed below
However answered, can't get all 6 unless gained mark for each of scheduling & polling]

- More than one job is in memory at same time
- More than one job is processed (apparently) at same time
- Real-time clock causes regular interrupts to create time-slices
- Scheduling allocates time-slices to each job
- Polling is the sequential checking of jobs so that each gets its appropriate share of time
- OS uses partitioning, ie the division of computer memory for different jobs
- OS pages jobs in and out to make better use of memory
- OS promotes efficient use of CPU

21

<lowercaseletter>	::=	a|b|c|……|z		)
<uppercaseletter>	::=	A|B|C|.....|Z		)1
<digit>	::=	0|1|2|……|9	(Condone if 0 missing)	)
<underscore>	::=	_		)
<char>	::=	<lowercaseletter>|<uppercaseletter>|<digit>|<underscore>		1
<chars>	::=	<char>|<char><chars>		1
<variablename>	::=	<uppercaseletter>|<uppercaseletter><chars>		1

[Marking: One mark for attempted recursion even if incorrect
(same item L and R + other item(s) on R are needed)
Max of 1 mark lost for notation
Other equally valid answers exist
However it's done, can only get 4 if completely correct]

---

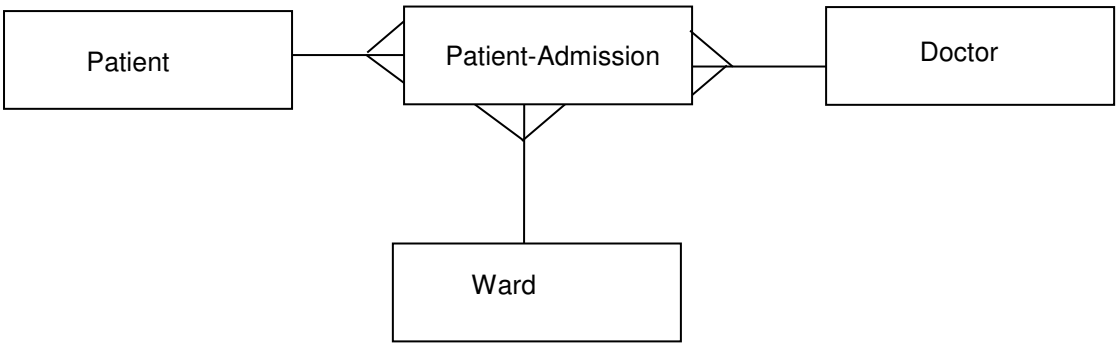
22 *Third normal form* means that data items are dependent on the whole key and nothing except the key (or the key, the whole key and nothing but the key). (**Idea of whole needed**) 1

***Or* (accepted not expected)** Third normal form is achieved when relations (tables) contain no partial or transitive dependencies.

Any 3 of:
- Normalising data usually reduces data duplication/redundancy
- Avoids danger of inconsistency / maintains integrity 3x1
- Avoids danger of data being lost during update
- Avoids wasting storage space / processing time

---

23 [**Marking**: Diagram 3 marks - each error minus 1 mark] 3



---

24 One approach (surrogate key)

PATIENT (PatCode, PatName, PatDoB)

WARD (WardName, NumBeds)

DOCTOR (DocCode, DocName, DocPager)

PATIENT-ADMISSION (AdmCode, PatCode, WardName, DocCode, AdmDate)

Alternative approach (non-surrogate key)

PATIENT (PatCode, PatName, PatDoB)

WARD (WardName, NumBeds)

DOCTOR (DocCode, DocName, DocPager)

PATIENT-ADMISSION (PatCode, AdmDate, WardName, DocCode)

[**Marking**: Four suitably named tables: 1
Each table with suitable Primary Key shown (2 or 3 correct = 1 mark) 2
One mark for each foreign keys (don't need to be identified as such) 3
-1 mark for any number of additional bad FKs
Ignore extra irrelevant fields etc]

25

```
1     declare Reading array(1..999) of integer (or real)
2     set Total = 0
3     set Max = 0 (or any integer < 0)
4     set NumHighs = 0
5     input NumReadings
6     for Count = 1 to NumReadings
7         input Reading(Count)
8         set Total = Total + Reading(Count)
9         if Reading(Count) > Max then set Max = Reading(Count)
10    endfor
11    set Mean = Total / NumReadings
12    for Count = 1 to NumReadings
13        if Reading(Count) > Mean then set NumHighs =  NumHighs+1
14    endfor
15    output “Mean Reading = ”, Mean
16    output “Number of readings above mean = ”, NumHighs
17    output “Highest  reading = ”, Max
```

Marking	
Initialise and first input	1
Loop structure	1
Input and two updates	1
Loop structure	1
Update	1
Calculate mean and three outputs	1

[**Marking**: Other approaches are possible and will be given full credit if correct.
No marks are given for brevity/efficiency/elegance]

26

- A CASE tool is a software tool which provides a number of functions which assist with the design and testing of a computer system / program.
- provides a data dictionary
- includes a graphics / diagram production feature
- may provide repositories of reusable code
- may provide project management tool(s)
- may incorporate version control
- may carry out report generation
- may include prototyping tool **- accepted but not expected**

- An application generator is a software tool which is used to assist and speed-up the creation of a (complete) system.
- provides a complete environment to support the programmer
- may include source code generator, compiler, etc
- usually creates code automatically
- may include an interface generator
- may include a debugger
- may be part of an integrated development environment **- accepted not expected**

- A compiler is a software tool which is used to translate a program (written in a high-level language) into a low-level program ready for execution on the computer.
- It has various stages: lexical analysis, syntax analysis, semantic analysis, code generation, optimisation (**1 mark for just naming at least 3**)
- During Lexical Analysis, input stream is broken into tokens, spaces etc, removed
- During Syntax Analysis, symbol table is produced, tokens are checked for fit to grammar
- During Semantic Analysis, a check is made that all variables are declared, and operations are legal eg real values are not being assigned to integer variables
- During Code Generation, machine code is generated
- During Optimisation, the code is improved if necessary to make it more efficient / faster / less resource greedy
- Produces error messages at any stage when needed (**once only for complier**)

- A debugger is a software tool used to detect, locate and correct faults in a program
- program trace/step-through/step-into: enables the programmer to see the progress through the program - which statements/procedures are being executed at any time
- break point: allows the programmer temporarily to halt execution in order to ascertain the value of variables at that point (or to step through the program from that point)
- variable watch: lists the value of a variable at specific points during the execution
- store dump: lists the entire contents of memory at a specific point
- error diagnostics: provision of messages relating to errors in the program
- post-mortem routines: enables programmer to see the values of variables at the point where the program failed - **accepted not expected**

Note: if only **name** various facilities e.g. trace, break point, variable watch, worth 1 only 10

[**Marking**: The description of any point can be extended with more detail to gain extra marks]

8-10 Candidates give a clear, coherent answer fully and accurately describing and explaining all areas. They use appropriate terminology and accurate spelling, punctuation and grammar.

5-7 Candidates describing and explaining a range of at least two of the areas, but responses lack clarity. There are a few errors in spelling, punctuation and grammar.

0-4 Candidates simply list a range of points or give a brief explanation of one of the areas. The response lacks clarity and there are significant errors in spelling, punctuation and grammar.

[Note: Max of 8 if all only 3 of the 4 sections attempted
Max of 6 if only 2 sections attempted
Max of 4 if only 1 section attempted]

WJEC
245 Western Avenue
Cardiff CF5 2YX
Tel No 029 2026 5000
Fax 029 2057 5994
E-mail: exams@wjec.co.uk
website: www.wjec.co.uk