Version 1.0: 0707

General Certificate of Education

Computing 6510

CPT4 Processing and Programming Techniques

Mark Scheme

2007 examination - June series

Mark schemes are prepared by the Principal Examiner and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation meeting attended by all examiners and is the scheme which was used by them in this examination. The standardisation meeting ensures that the mark scheme covers the candidates' responses to questions and that every examiner understands and applies it in the same correct way. As preparation for the standardisation meeting each examiner analyses a number of candidates' scripts: alternative answers not already covered by the mark scheme are discussed at the meeting and legislated for. If, after this meeting, examiners encounter unusual answers which have not been discussed at the meeting they are required to refer these to the Principal Examiner.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of candidates' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

Set and published by the Assessment and Qualifications Alliance.

The Assessment and Qualifications Alliance (AQA) is a company limited by guarantee registered in England and Wales (company number 3644723) and a registered charity (registered charity number 1073334).
Registered address: AQA, Devas Street, Manchester M15 6EX
Dr Michael Cresswell Director General

Instructions to examiners

The following forms of notation should be used on candidates' scripts:

- Ticks - To indicate what is accepted as correct or creditworthy, placed in the body of the answer, and on diagrams;
- Underscoring - To identify errors/irrelevance in written answers;
- Crosses - to indicate a wrong answer;
- Brief comments - placed in at suitable points in the body of the text to amplify the marking;
- BOD - means benefit of the doubt and is used where the candidate's answer has been given a mark on the balance of probabilities that the candidate's answer has met the requirements of the mark scheme even though it could be interpreted differently;
- NE - means not enough and is applied to an answer that falls short of what is required;
- O/S - means outside the mark scheme. The candidate's answer is creditworthy but the answer does not match any of the answers on the mark scheme for the particular question. Nevertheless a mark is awarded;
- C/F - means carried forward. This arises when a candidate offers an answer that is not creditworthy in that part of the question but is creditworthy in a later part of the same question. The mark is carried forward to the part of the question that is creditworthy;
- C/B - means carried back. This is similar to a carry forward but the mark is carried back to an earlier part of the question.
- T/O - means talked out. The candidate's answer is contradictory.
- F/T - means followed through. If the candidate made a mistake in the earlier part of an answer, mark the answer using the correct method on their answer from the earlier part.

The following notation is used in the mark scheme

- ; - means a single mark;
- **A** - means an acceptable creditworthy answer;
- **R** - means reject answer as not creditworthy.
- **I** - ignore
- / - means alternative word or sub-phrase
- / / - means alternative answer

General Rules for marking
Ignore Abbreviations
Ignore Brand Names

1	(a)		35,37;	1	
	(b)		53,55;	1	
	(c)		00111001;; 1 mark for sign, 1 mark for value **A** 11000111 1 mark	2	
	(d)		0.1110010 ; 00000110;	2	
	(e)		To allow a wider range of values to be stored; To minimise rounding errors// greater accuracy/precision using a <u>given number of bits</u>;	2	
					8
2	(a)	(i)	To provide a standard interface to the O.S.; **A** Communicates with the O.S.; To control the hardware/device; To sense the status of the device; To read/write registers in the device controller;	max 2	
		(ii)	To allow a range of hardware; That was not available when the O.S. was written; To allow compatibility between different manufacturers; To reduce the size of the O.S.;	max 2	
		(iii)	A language that allows registers to be read/written// Assembly Code; **A** Machine Code **A** 'C'/BCPL/CPL **A** Low level language	1	
		(iv)	Requires direct access to the registers/memory (in the I/O (device) controller); **R** access hardware Needs high speed execution/processing; Uses minimum <u>memory</u>// Minimises program code; Needs to allow low level commands found in the instruction set; **A** Assembly Code/'C'/BCPL/CPL easier to write than machine code if consistent with (iii)	max 2	
	(b)	(i)	Mouse click // mouse movement // wheel movement;	1	
		(ii)	Disable interrupts; <u>Save</u> Volatile Environment; **A** Registers Branch to 2nd level interrupt handler; Obtain parameters (from stack); Transfer/process data; **A** Deal with interrupt <u>Restore</u> Volatile Environment; **A** Registers Enable interrupts; Return control;	max 3	
					11

3	(a)	To perform mathematical operations/calculations;	1	
	(b)	Fetches, decodes and executes instructions; To control/co-ordinate the other parts of the processor;	max 1	
	(c)	Accumulator; Index Register; Base Register; Stack Pointer; Current Instruction Register; **A** Instruction Register Program Counter/ Instruction Pointer/ Sequence Control Register; Memory Address Register; Memory Data Register// Memory Buffer Register; Flags Register// Status Register// Condition Code Register; Interrupt Register;	max 3	**5**
4	(a)	Surface/layer number; Track/Cylinder Number; Sector/Block Number;	3	
	(b)	Memory; used for temporary storage of one or more file blocks; in transit between disk and main memory;	max 2	**5**
5	(a)	person (ronald); meat (chicken); eats (ronald,chicken) eats (ronald,lamb); **I** Order	3	
	(b)	sarah; john;	2	
	(c)	NOT;(eats(Name,Food); AND meat(Food)); **I** person(Name) **I** vegetable(Food) **I** Order **R** eats(Name,potato) OR eats(Name,carrot) 3 Penalise case once only in (a) and (b)	max 3	**8**

6

Label	Opcode	Operand	Comment
Start	LD	Temp;	
	CMP	#0A;	**A** #10
	JGE	Label1;	
	LD	House	Load and store
	AND	#FE;	House;
	ST	House	
	JP	End	
Label1	LD	House	Load and store
	OR	#01;	House if not
	ST	House	Given above;
End	JP	Start	

Or

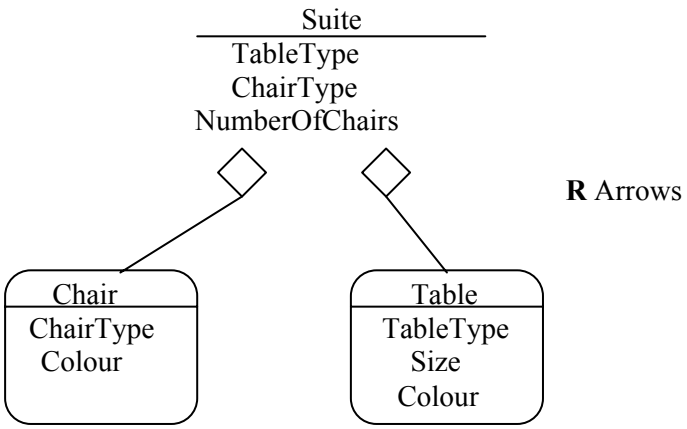
Label	Opcode	Operand	Comment
Start	LD	#0A;	**A** #10
	CMP	Temp;	
	JL	Label1;	
	LD	House	Load and store
	AND	#FE;	House;
	ST	House	
	JP	End	
Label1	LD	House	Load and store
	OR	#01;	House if not
	ST	House	Given above;
End	JP	Start	

1 mark for working solution
R Storing 0 and/or 1 in House
Penalise use of each wrong number base once only 7

7

7 (a) An object that contains other objects;
A A class containing other classes; 1

(b) (i)



R Arrows

1 mark for class entries
1 mark for connections **A** circles or diamonds, filled or not 2

(ii)

```
Chair = class;
   Private;
      ChairType : string/text   A integer      }
      Colour : string/text    A integer/color  } ;

Table = class;
   Private                                      }
      TableType : string/text A integer         } ;
      Size : string/text A integer              }
      Colour : string/text A integer/color      }

Suite = class;
  (Public)
     (procedure) DisplayDetails;
   Private                        }
      TableType : Table           }
      ChairType : Chair           } ;
      NumberOfChairs : integer;
```

A any sensible syntax
R implied inheritance

max 8

11

8 (a) A procedure that is defined in terms of itself;
A A procedure that calls itself
R re-entrant 1

(b) Store return addresses;
Store parameters; max
Store local variables/ return values; 1

(c)

Number	Entry	Output
11	1	
11	2;	
11	3;	
11	4;	4;

4

(d) A linear search//
To find/output the position/index of Number in Items; 1

(e) Number is not an entry in Items// Stack overflows; 1

(f) Test for reaching the end of Items; 1

(g) Binary Search; max
An iterative solution; 1

10