2010 HSC Software Design and Development Sample Answers

This document contains 'sample answers', or, in the case of some questions, 'answers could include'. These are developed by the examination committee for two purposes. The committee does this:

(a) as part of the development of the examination paper to ensure the questions will effectively assess students' knowledge and skills, and

(b) in order to provide some advice to the Supervisor of Marking about the nature and scope of the responses expected of students.

The 'sample answers' or similar advice are not intended to be exemplary or even complete answers or responses. As they are part of the examination committee's 'working document', they may contain typographical errors, omissions, or only some of the possible correct answers.

Section II

Question 21 (a)

Sample answer:

Users provide insight into all stages of the process in particular the following areas:
- Specifications
- Screen design
- Testing
- Social and technical issues including ergonomics

Question 21 (b)

Answers could include:

- description of modular approach
- benefits of the modular approach include:
  - helps understanding of complex solution by focusing on smaller problems
  - facilitates team work by allowing different groups to work on smaller modules
  - facilitates planning of complex solution
  - aids testing by allowing individual modules to be tested and debugged
  - makes the re-use of code or use of existing/library routines easier
  - makes modification/maintenance of a complex solution easier by enabling focus on smaller logical units.

Question 21 (c)(i)

Answers could include:

- Saves development time
- Routines are already tested hence saves testing time
- Helps maintenance by providing well-understood standard routine with consistent behaviour
- Library routines can be updated independently
- Library routines can be re-used by a variety of different providers without modification

Question 21 (c)(ii)

Answers could include:

- Deciding whether the library routine is appropriate for a range of production uses
- Consider the parameters that need to be passed
- Need for thorough testing under a range of conditions
- Thorough documentation of the purpose and expected inputs and outputs of the routine.

Question 21 (d)(i)

Sample answer:

Technical documentation is not critical in this case. An end user approach is not appropriate, as this game is not intended for just the developer's use.
A structured approach is not appropriate because specifications are not clear and delays in development would not allow for good involvement of the friends.

- RAD is appropriate because the developer has had some experience in writing code
- Scheduling is flexible – no time constraints
- Low budget
- Allows quick development of versions that friends can use and evaluate
- Prototyping is appropriate because:
    - requirements are not well defined and the gamer has had no experience in writing games
    - the input from her friends is made easier by using a prototype rather than more formal descriptions
- Prototyping is well suited to an interactive screen-based application

Question 21 (d)(ii)

Answers could include:

Forms of documentation could include:
- user documentation
- self documentation of code
- documentation for maintenance of code

Internal/intrinsic documentation in the code would be important to help the developer in making changes at a later stage.
Online documentation eg help would benefit the gamers better than printed user guide.
A tutorial or demonstration would help in introducing software to a user.

Question 21 (d)(iii)

Answers could include:

Event-driven approach – driven by the user, path is determined by the user's interactions
Sequential approach – logic and path through the program is determined by the programmer

- The event-driven approach is better suited to games with a high level of user interactivity and choice eg interactive screen-based game
- Sequential approach would mean that the programmer provides a limited number of choices for the user at any stage
- The choice of languages used by the gamer will influence the ease of implementation of each type of game

Question 22 (a)

Answers could include:

1. Need to consider the inputs to the algorithm and expected outputs from the algorithm
2. Need to consider the logic of the algorithm and choose combinations of input values which test each logical path through the algorithm
3. Need to choose input values which test boundary conditions
4. Need to consider exceptional values such as out of range or incorrect data types. Numbers such as zero or negative numbers should also be considered.

Question 22 (b)(i)

Sample answer:

```
BEGIN SUB PROGRAM
String match (Search string, data string)
n= length of search string
Output = "Same"
Index = 1
While output = "Same" and Index < n
          IF search string (index)<>"?" THEN
                  IF Search String(index)<> data String(index) THEN
                        OUTPUT = "Different"
                  ENDIF
          ENDIF
      Index = Index + 1
ENDWHILE
PRINT OUTPUT
ENDSUBPROGRAM StringMatch
```

Question 22 (b)(ii)

Answers could include:

There are two situations to be considered:
1. The loop terminates early and there would be a partial match eg fish and fishcake would incorrectly match given an incorrect output
2. The index could be out of range resulting in a run-time error when comparing elements of the two strings

The student could solve this problem by error trapping the length of the two strings.
Comparison of elements should only take place if the strings are of equal lengths.
If the lengths are different the strings don't match so the output should be "Different"
(assuming spaces are treated the same as any of the characters).

Question 22 (c)(i)

Sample answer:

After the first statement array[index A] takes on the value of array[index B] so the sound statement assigns the value of array[index B] to array[index B] so both elements have the same value – there is no swap.

To correct this algorithm the code should be:
Temp = array[index A]
Array[index A] = array [index B]
Array[index B] = temp

Question 22 (c)(ii)

Answers could include:

Note: A diagram may be used to illustrate part of the logic of this process.

The list is divided into a sorted part and an unsorted part. Initially there will be one element in the sorted part (either the first or the last). Each element in the unsorted part is placed in the sorted part using a linear search to find its correct position. The sorted part increases and the unsorted part decreases until the whole list is sorted.

Question 22 (d)

Answers could include:

Techniques
Through documentation both intrinsic and extrinsic
Meaningful variable names
Comments
Indented well laid out code
Log of modifications
Use of standard control structures
Use of a modular approach – one logical task per subroutine
Use of library routines

Discussion Point
Appropriate use of constraints
These techniques may be discussed in terms of:

• understanding the code

• allowing modifications to some code without affecting other parts of code, limiting changes that need to be applied

• making software bug-free and hence reducing need for maintenance

Question 23 (a)

Answers could include:

Gantt chart
- facilitates scheduling and coordination of tasks
- allocation of resource and personnel

Log book
- facilitates communication of progress among team members
- allows review and evaluation of project and how well team work together

Question 23 (b)

Answers could include:

- At the module level, test the syntax of the code in the module, define criteria and test inputs for the module, test the module with the chosen test inputs and check whether the module produces expected results.
- At the program level, assuming a program comprises a collection of modules, check the code that calls different modules (eg by introducing breakpoints print statements to see if flow (to/from) between modules is correct. Identify test inputs for the program and check whether the program produces expected results.
- Test the interactions between programs and modules and where programs/modules are called and returned to test the flow.
- At the system level, identify test inputs for the overall system and check outputs for these.
- Testing the interaction between different programs. Check hardware, software, procedures, people and data work together.

Given a large system often involves many modules and programs and in a project, development of these modules/programs can be done by different members of a team in testing the software solution, communication and interaction between these different developers is necessary (to ensure correct defining of test inputs and checking outputs as well as testing the flow of the overall system).

Question 23 (c)(i)

Sample answer:

Assuming the instruction is stored in memory the CPU fetches the instruction from the memory using the program counter pointing to memory location via address bus and data bus. The processor decodes LOAD and fetches the content of memory A (via data bus) and stores the contents from the bus to the register R1.

Question 23 (c)(ii)

Sample answer:

The ADD instruction would execute faster as it adds the contents of two registers in the CPU and puts the result back in another register within the CPU. All the operations involve only registers within CPU – fastest.
Other instructions involve accessing memory locations which are in general slow and involve communications via buses (outside of CPU) to memory.

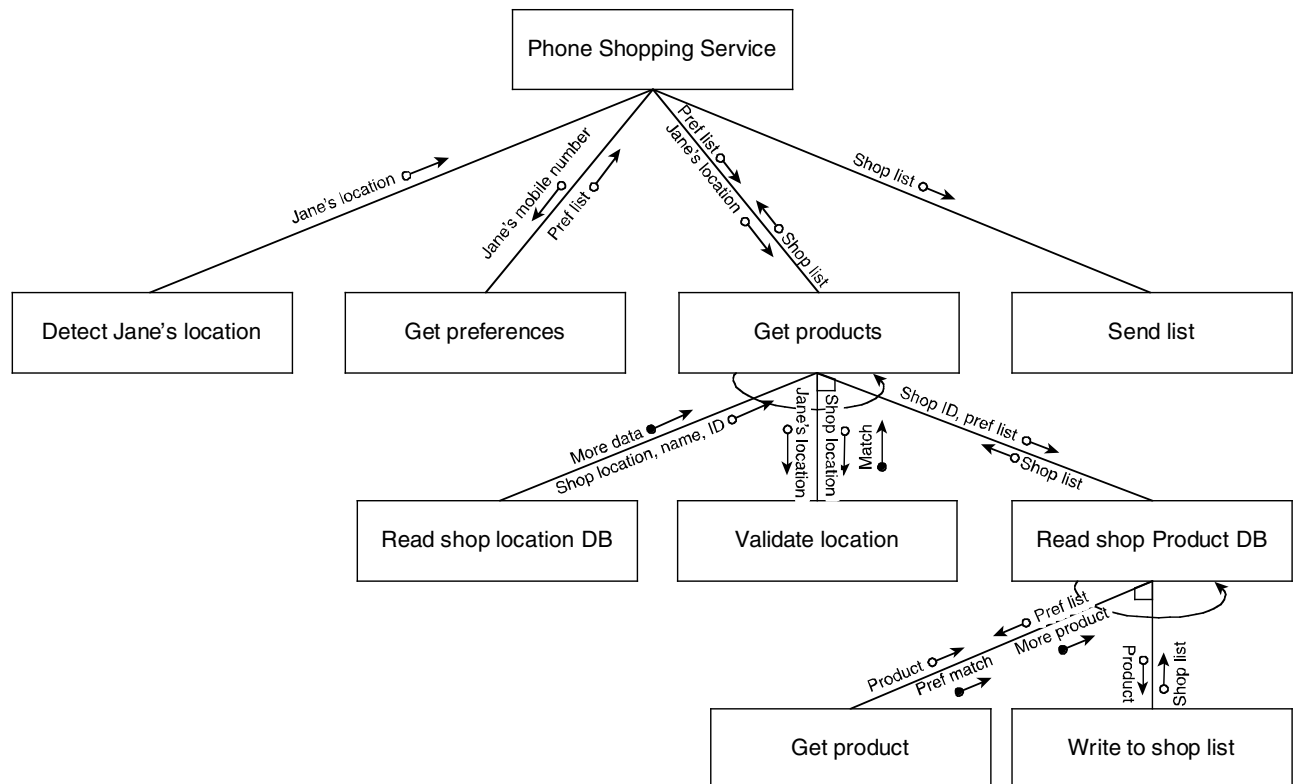**Question 23 (d)(i)**

Sample answer:

Storing price range data as two separate numeric values. Storing as two separate values means that comparisons can be made on maximum and minimum values, as such comparison is possible and the current situation. Storing as numeric data means calculations and comparisons can be made without having to extract the numeric values from string data.

Question 23 (d)(ii)

Answers could include:

- Use a consistent and smaller font for readability
- Provide a means for data input
- Provide text box for brand so that it can be edited
- Redesign screen so that information fits without scrolling
- Provide a meaningful heading to reflect the category chosen "shoes"
- Provide navigation buttons to return to screen 1
- Separate out viewing and editing with screen 2 for viewing and screen 1 for editing, possibly with different screen for each item. This will allow a clearer display.

Question 23 (d)(iii)

Sample answer:

Answers could include:

Main logic focused on Jane's location.

Section III

Question 24 (a)(i)

Answers could include:

The basic building blocks of the logic paradigm are:
Facts – for example orbits (mars, sun) and
Rules – for example planet CP orbits CP, sun

Question 24 (a)(ii)

Answers could include:

The following values would be returned for X:
Mercury
Venus
Earth
Mars

Question 24 (a)(iii)

Answers could include:

Backward chaining will start with the assumption that phobos is a satellite and then work backwards to find rules and facts in the knowledge base to prove its validity.
It will start with satellite (S) $\Leftarrow$ orbits (S,P) and planet (P) and establishes that mercury, venus, earth and mars are planets in that orbit (phobos, mars), hence phobos is a satellite.

Question 24 (a)(iv)

Sample answer/Answers could include:

An advantage of the logic paradigm in this case is that new rules and facts can easily be added as they are discovered. In the imperative approach, the program itself would have to be modified and re-compiled.
The logic paradigm enables new inferences to be made from the data, the imperative approach does not allow new inferences to be made. In the imperative approach every inference has to be coded.

Question 24 (b)(i)

Sample answer:

CLASS WritingImplement
ATTRIBUTES colour =
             Brand =
             Price =
             Medium (link/lead/paint)

METHODS:
             Write
             Erase
ENDCLASS

Note: A written response to this question is equally valid.

Question 24 (b)(ii)

Answers could include:

CLASS Pen
      Is a subclass WritingImplement
ATTRIBUTES: Style
             Thickness
ENDCLASS

Note: a written response to this question is equally valid.

Question 24 (b)(iii)

Answers could include:

Because Pencil is a subclass of WritingImplement it inherits all of the ATTRIBUTES and METHODS of WritingImplement and so can be used as it stands as a class.

Polymorphism allows the class Pencil to override the inherited behaviour of Erase and to use its own method for erasing.

Question 24 (b)(iv)

Answers could include:

Encapsulation is the isolation of an object (in this case Pen) from its environment, so that changes to the object (such as changing the Write method in this case) can be made without affecting other parts of the system.
As technologies change, individual objects can be modified in this way. Knowing that these changes are shielded from the rest of the system, as long as the interface to that object remains the same.

Question 25 (a)(i)

Sample answer:

2	2^0	2^{-1}	2^{-2}
1	0	1	1

$$1 \times 2 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$
$$= 2 + 0 + \frac{1}{2} + \frac{1}{4}$$
$$= 2\frac{3}{4}$$

Question 25 (a)(ii)

Sample answer:

$$\begin{array}{r} 10.11 \\ +10.1 \\ \hline 101.01 \end{array}$$

Question 25 (a)(iii)

Sample answer:

A half adder takes two simple-bit inputs and outputs a simple-bit sum and simple-bit carry. A full adder takes a two simple-bit inputs and a simple bit carry-in and outputs a simple-bit sum and a simple-bit carry

Question 25 (a)(iv)

Sample answer:

Many fractions will be represented by non-termination binary representation (not only $\frac{1}{10}$ but also $\frac{1}{3}, \frac{1}{5}$ etc). Floating point representation (using mantissa and endpoints) allows more accurate representation of fractional values. Precision of accuracy is improved in floating point representation for size of length of representation.
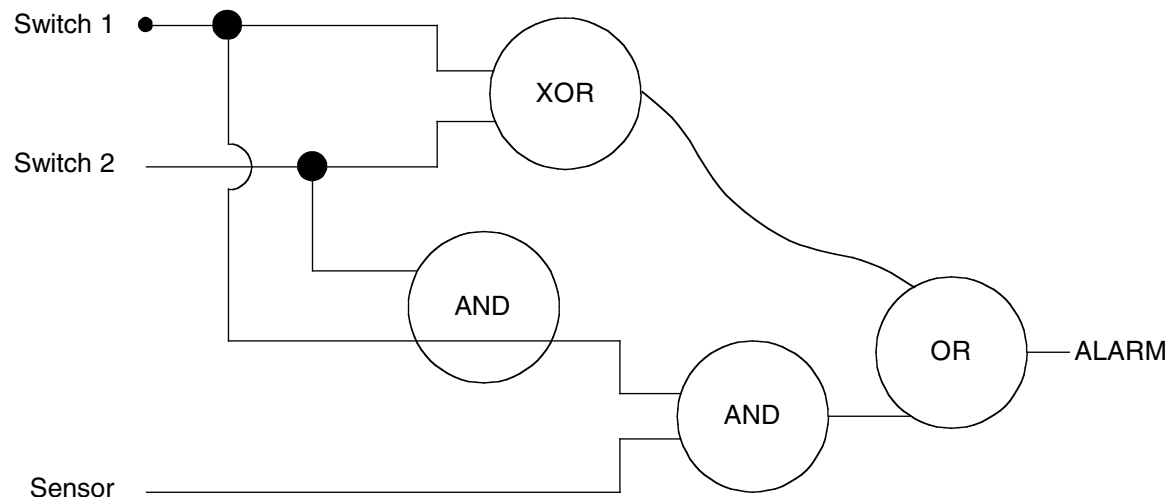
Question 25 (b)(i)

Sample answer:

| Input			Output
Switch 1	Switch 2	Sensor	Alarm
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Question 25 (b)(ii)

Sample answer:

Question 25 (b)(iii)

Sample answer:

The alarm is completely off to start with (all switches, sensor alarm are off) then the sensor is activated next the switch is turned on the sensor is still activated and the alarm sounds.

Question 25 (b)(iv)

Sample answer:

Inputs | Outputs

R	S	Q	Q̄	
1	1	–	–	
1	0	0	1	**Useful**
0	1	1	0	**States**
0	0	–	–	
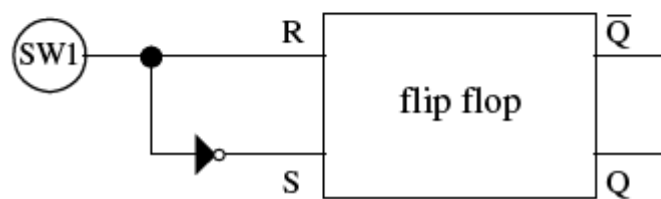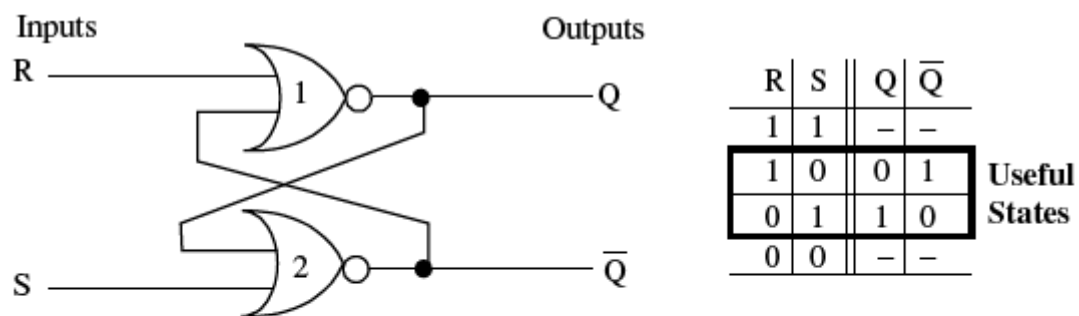
Four flip flops needed, one for each device (switch, sensors, alarm)