

2012 HSC Software Design and Development 'Sample Answers'

When examination committees develop questions for the examination, they may write 'sample answers' or, in the case of some questions, 'answers could include'. The committees do this to ensure that the questions will effectively assess students' knowledge and skills.

This material is also provided to the Supervisor of Marking, to give some guidance about the nature and scope of the responses the committee expected students would produce. How sample answers are used at marking centres varies. Sample answers may be used extensively and even modified at the marking centre OR they may be considered only briefly at the beginning of marking. In a few cases, the sample answers may not be used at all at marking.

The Board publishes this information to assist in understanding how the marking guidelines were implemented.

The 'sample answers' or similar advice contained in this document 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

Sample answer:

Pilot implementation involves a select group using the entire new system for a trial period. When the new system proves to be adequate, the rest of the company transfers across to it. For example, a supermarket chain wants to implement a new cash register system. They implement it in one store first and when it proves successful, it is implemented in all stores.

In phased implementation the new system is introduced to the entire organisation module by module, until the new system has been fully implemented. For example a bank has a new system to implement. They begin by installing the module that deals with new deposits in all branches of the bank, then they install the withdrawals module, then the loans module, until all modules have been installed.

Answers could include:

Pilot: users as part of the trial period could be used to educate and train the rest of the employees.

Phased: the gradual/staged implementation of a system within an organisation. Resource constraints are a potential reason why a phased approach needs to be used.

Question 22 (a)**Sample answer:**

Changes could include improvement to variable names, eg replace 'x' and 'y' with 'friend1' and 'friend2' and replace 'i' and 'j' with 'friend1Location' and 'friend2Location'.

Add indenting, eg indent the four inner lines of subroutine BeFriend.

Add comments for each LinearSearch and/or a comment above the two assignment statements.

Question 22 (b)**Sample answer:**

personNo	friend
333	555
348	555
428	348
555	348

Question 22 (c)**Sample answer:**

The subprogram does not cater for the storage of more than one friend, as the website specifications require. Each new friend overwrites the previous friend. A more sophisticated data structure will be more suitable for the requirements.

The linear search is inefficient as there will be many thousands of records and the scenario states that the records are ordered by personNo. A more efficient search method would be the binary search.

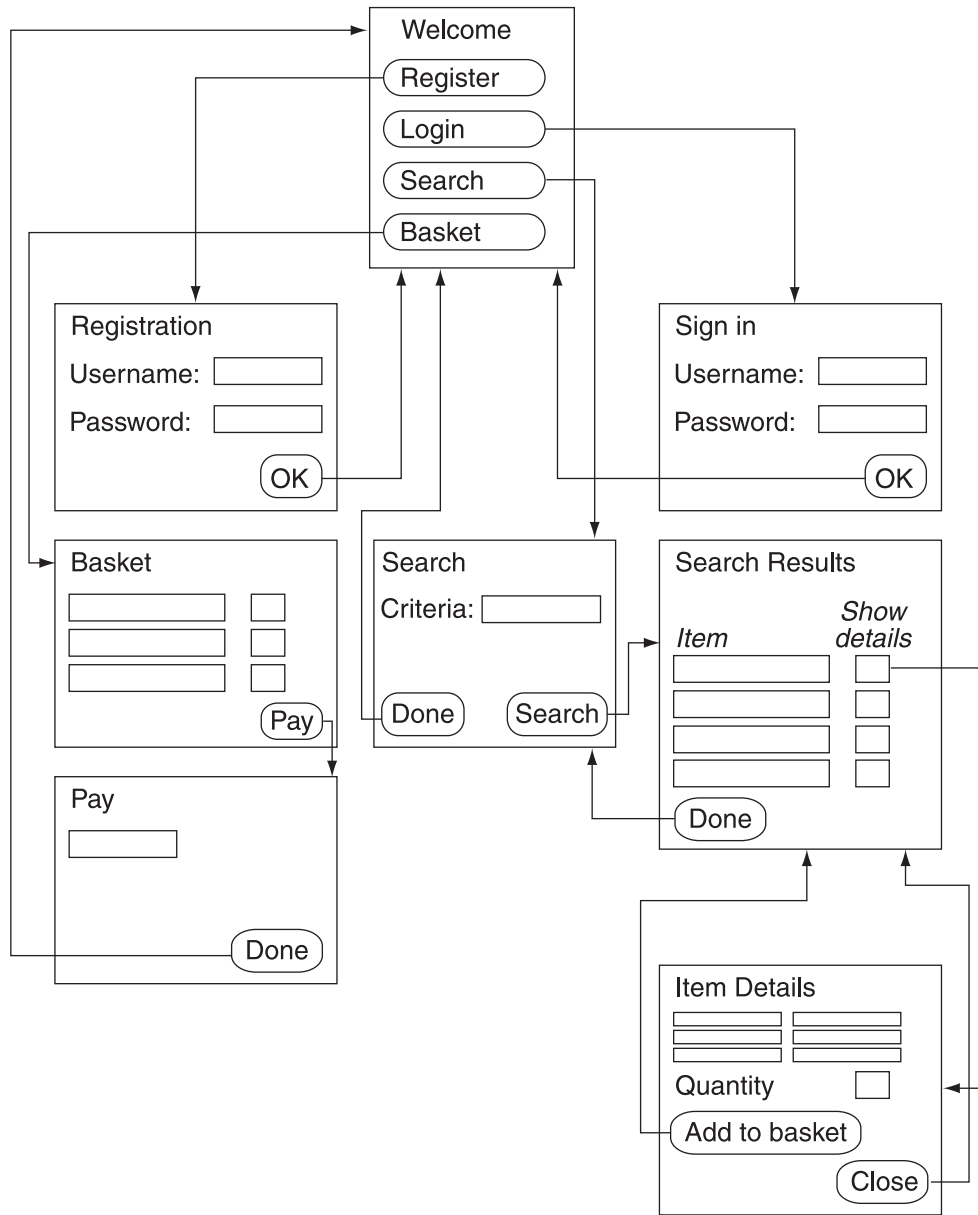
Answers could include:

The code assumes that people with personNos 'x' and 'y' will be found in the array – an 'if' statement should be added to handle that contingency.

It is not clear how to indicate if a person has no friends. There should be a special value, such as 0 or -1, to indicate this.

Question 23 (a)

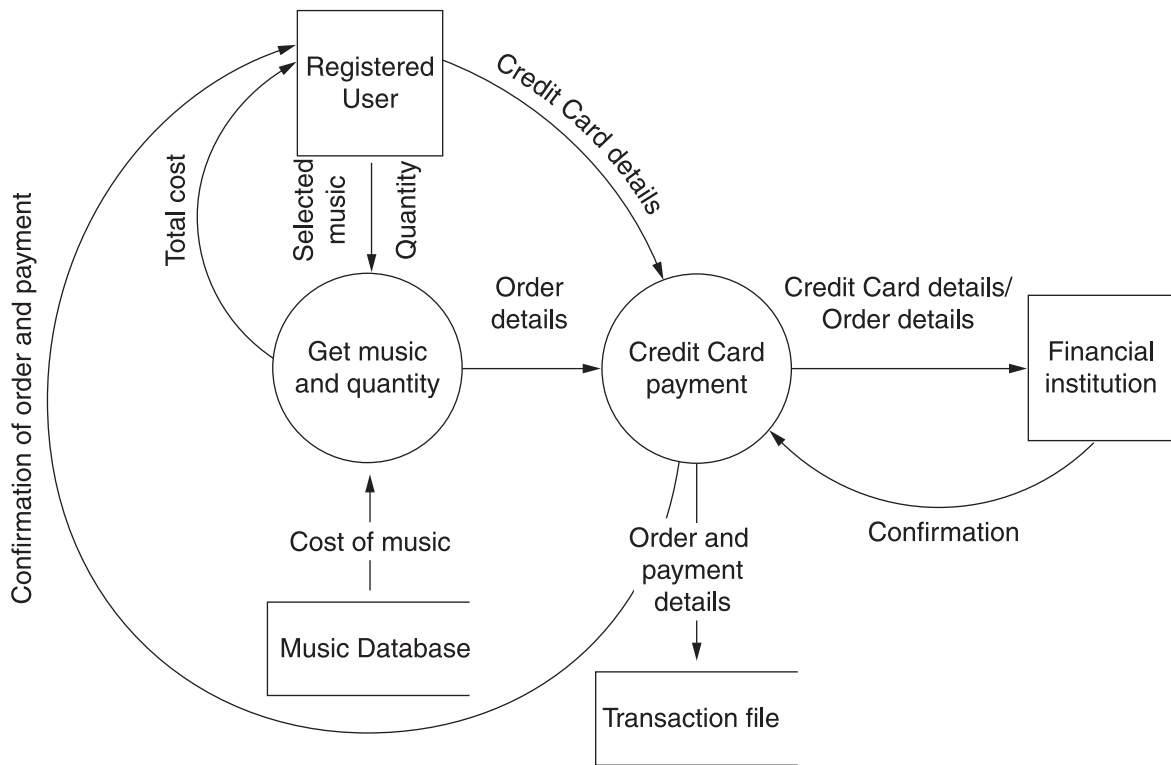
Sample answer:



Note: Every screen may have a **Cancel** button that takes the user back to the main screen.

Question 23 (b)

Sample answer:



Question 24***Sample answer:***

```
BEGIN BalancePlane
    LeftCount = 0
    RightCount = 0
    FOR Row = 1 to 50
        FOR Column = 1 to 6
            IF SeatArray (Row, Column) <> 0 THEN
                IF Column < 4 THEN
                    LeftCount = LeftCount + 1
                ELSE
                    RightCount = RightCount + 1
                END IF
            END IF
        NEXT Column
    NEXT Row
    IF RightCount > LeftCount THEN
        Print 'Right side too many'
    ELSE
        IF RightCount < LeftCount THEN
            Print 'Left side too many'
        ELSE
            Print 'Balanced'
        END IF
    END IF
END BalancePlane
```

Question 25 (a)***Sample answer:***

An agile approach is suitable because it caters for:

- the speed of getting solution to market
- an interactive approach with selective refinement
- a working version delivered after each iteration
- a good response to changing specifications
- close collaboration between development team and client throughout the development process.

Question 25 (b)***Sample answer:***

Some functionality requirements are:

- wide range of devices
- fast response
- full-colour display
- voice-live
- video
- available 24/7 and worldwide
- accept enquiries
- multilingual
- search engine.

Some compatibility issues are:

- different operating systems
- different web browsers
- different devices
- different screen sizes
- storage capabilities
- network capabilities
- different language versions
- different time zones.

Some performance issues are:

- transmitting and processing (video and voice)
- traffic and peak loads
- quality of display
- time differences.

Question 26 (a)***Sample answer:***

When employing a bottom-up approach where a developer needs to test a lower-level module before a higher-level module is completed, a driver may be used to provide test data.

Answers could also include:

- A driver can be used to provide data when a large volume of data needs to be entered. In this case, the data can be read from a file.
- A driver may be used to replace a random number generator within a program. This would enable the input of known values into subsequent processes and thus assist in the testing of the program.

Question 26 (b)***Sample answer:***

Test data for this scenario would need to be in sets. Each set would provide matching input values for destination and weight as well as corresponding expected values for the acceptance message and the cost.

Values for destinations should include all possible states as well as unanticipated inputs. Values for weights should provide both boundary and path testing values, relative to the destination, as well as unexpected values.

Question 26 (c)***Sample answer:***

```
BEGIN TestDriver
    Open TestData file for input
    Open ErrorFile file for output
    Read TDest, TWght, TMsg, TCost from TestData file
    WHILE TDest is not EOF(TestData)
        Quotation (TDest, TWght, msg, cost)
        IF msg <> TMsg OR cost <> TCost THEN
            Write TDest, TWght, TMsg, TCost to ErrorFile
        END IF
    Read TDest, TWght, TMsg, TCost from TestData file
    ENDWHILE
    Close TestData file
    Close ErrorFile file
END Test_Driver
```


Question 27 (a)***Sample answer:***

One advantage is that you will not lose your data even if your local device (eg your laptop or tablet) is damaged or lost, as the data is stored at another location, separate from your device.

Nonetheless, since the data is stored remotely, other people may be able to access the data causing security issues. When the data is accessed via shared computers, the data may also be retrieved by other people if no encryption or security measure is implemented.

Answers could include:

Advantages

- Data is available from anywhere
- Potentially unlimited storage
- Easy sharing of data
- Scalability

Disadvantages

- Privacy
- Ownership
- Loss of control and/or access
- May not work behind firewalls

Question 27 (b)***Sample answer:***

Developers should consider minimising the local storage size of the cloud computing application so that it can be stored and used on a wide range of devices. Speed of accessing the data is another important factor as the data will be stored remotely. While the data can be accessed anywhere any time, it is important that the response time is made as short as possible.

Answers could include:

- Accessibility for multiple platforms or browsers
- The importance of the user interface in streamlining repetitive tasks such as uploading lots of files
- Reliability
- Security

Question 28 (a)***Sample answer:***

```
Let temp = x(1)
FOR i = 1 to length-1
    x(i) = x(i+1)
NEXT i
x(length) = temp
```

OR

```
Let Temp = x(1)
FOR i = 2 to Length
    x(i-1) = x(i)
NEXT i
x(Length) = Temp
```

Note: a variable name other than 'i' is acceptable.

Question 28 (b)***Sample answer:***

```
FOR i = 1 TO MLENGTH
  IF MESSAGE(i) <> " " THEN
    alphaindex = 0
    Found = false
    WHILE Found = false AND alphaindex < 26
      alphaindex = alphaindex + 1
      IF ALPHABET (alphaindex) = MESSAGE (i)
        Found = true
      ENDIF
    END WHILE
  ENDIF
  IF Found = false THEN
    MESSAGE (i) = "?"
  ELSE
    IF alphaindex < 25 THEN
      MESSAGE (i) = ALPHABET (alphaindex + 2)
    ELSE
      MESSAGE (i) = ALPHABET (alphaindex + 2 - 26)
    ENDIF
  ENDIF
ENDIF
NEXT
```

Features to look for:

- evidence of going through each character in the message
- searching for each character in the alphabet
- replacing letter-by-letter 2 positions down in the alphabet
- leaving spaces as spaces
- encoding invalid character with ‘?’
- handling the letters ‘Y’ and ‘Z’ appropriately
- handling exiting conditions of loop(s) correctly.

Question 29***Sample answer:***

With the monthly reporting, it is difficult to provide effective communications. Further, emails may not be read or acted on promptly. They also do not allow for effective discussion. The team should meet more frequently (maybe once a week) via a video/voice conferencing tool that will facilitate conversation between project members.

Another risk is the installation of the software on the client's system before the user documentation has been written. If clients do not know how to operate the software it will be impossible to use successfully. A proposal to reduce this risk is to develop the documentation during development and give it to the user before or during installation of the new software.

Answers could include:

- No testing by the client in the development process
- Not enough communication between the team leader and the client, or between the team members
- Changing of specifications means time will be wasted on unnecessary work for a month before realising
- Developers may create modules that don't interface well and should use some form of cloud computing or online documents to create data dictionaries and other technical documentation

Question 30 (a)**Sample answer:**

The error is that the string is reconstructed in the wrong order.

The correct code is

```
IDString = IDString+IDcharacter
```

Question 30 (b)**Sample answer:**

```
BEGIN MAINPROGRAM
```

```
count = 0
```

```
WHILE monitoring device is on
```

```
    WHILE NOT BOF
```

```
        READ IDCharacter
```

```
    END WHILE
```

```
    IDString = “ ”
```

```
    count = count + 1
```

```
    CALL getID(IDString)
```

```
    IDArray(count)= IDString
```

```
ENDWHILE
```

```
END MAINPROGRAM
```

```
BEGIN MAINPROGRAM
```

```
count =0
```

```
REPEAT
```

```
    IDString = “ “
```

```
    READ IDCharacter
```

```
    IF IDCharacter = BOF THEN
```

```
        getID(IDString)
```

```
        count = count + 1
```

```
        IDArray(count) = IDString
```

```
    END IF
```

```
    Check for System Shutdown Command
```

```
UNTIL System Shutdown
```

```
END MAINPROGRAM
```

Question 30 (c)**Sample answer:**

There are a number of fields with different data types, for example deviceID is an integer, employeeID is a string and there are date/time fields. As different data types need to be stored and the data is related, a record structure is appropriate.

device ID	Date	Time	employee ID
-----------	------	------	-------------

Section III

Question 31 (a)

Sample answer:

A limitation of the imperative paradigm is its difficulty in solving certain types of problems. OO is more suitable than imperative for many gaming programs due to the use of classes and subclasses using inheritance, encapsulation and polymorphisms. Logic is more suited to AI programs due to the availability of facts, rules and an inference engine.

Another limitation is the need to specify code for every individual process. OO can use attributes and methods to cut down on large parts of the coding. Logic programs can use an inference engine to reduce the need for coding when searching through a database.

Answers could include:

- Difficulty of coding for variability and reusability
 - OO can make use of its objects and methods combined with a modular approach to make the code more variable and reusable

Question 31 (b) (i)

Sample answer:

Using backward chaining, the program would look at the rule, `player_in_team(Legends, Jess)`, then try to locate a fact, `player(Jess)`, which would be found and a fact, `in_team(Legends, Jess)`, which would not be found and infer that it is false.

OR

Using forward chaining, the program would search the facts to find `player(Jess)`, `in_team(Legends, Harley)` and `in_team(Rockets, Jess)` which could not be used to support `player_in_team(Legends, Jess)`: - `player(Jess)`, `in_team(Legends, Jess)` and would evaluate X to be false.

Question 31 (b) (ii)***Sample answer:***

```
player(Enoch)
in_team(Rockets, Enoch)
captain_of_team(Rockets, Enoch)
coach_of_player(X, Y):- coach_of_team(Z, X), player_in_team(Z, Y)
```

Question 31 (c) (i)***Sample answer:***

By this approach the variable balance can only be updated by a call to withdraw and that means that the balance will always be checked as part of every withdrawal.

Question 31 (c) (ii)***Sample answer:***

Deposit should be added to the public section of the class ACCOUNT from where it is inherited by all the subclasses.

Question 31 (c) (iii)***Sample answer:***

A method 'withdraw' should be added to the subclass 'BlueAccount'. This method will be expected instead of the 'withdraw' method in class ACCOUNT. When two classes have a method of the same name, as in this case, so that different things are done by the two methods, it is called polymorphism.

Question 31 (d)***Sample answer:***

The logic paradigm is not a good way of describing the sequential aspects of the computer game and requires a combination with another paradigm for a complete solution.

OO allows for the use of inheritance, where character types share properties and characteristics. A single parent class could implement the common properties and characteristics.

Using the logic paradigm, the determination of who wins a fight could be concisely formulated as a set of logical rules.

Disadvantage of OO code – some character types may need to inherit properties from more than one super class. Some languages (like Java) do not support multiple inheritance. While some languages (like C++) do support multiple inheritance, it can be difficult code to maintain.

Question 32 (a)**Sample answer:**

Q is the 17th letter of the alphabet.

The hexadecimal ASCII code for Q is $40 + 11 = 51$.

The hexadecimal ASCII code for the lower case q can be found by adding 20 to 51.

Question 32 (b)**Sample answer:**

A 4-bit system allows 16 combinations.

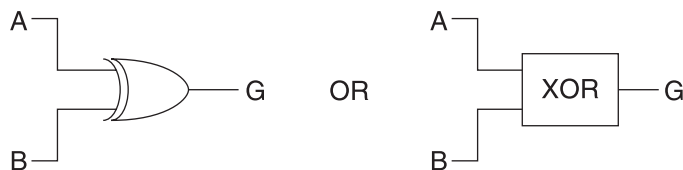
For sign and modulus, there are two zeros (+0 and -0), 7 positive numbers (1 to 7) and 7 negative numbers (-1 to -7). Hence 15 numbers in total are represented.

For two's complement, there is one zero, 7 positive numbers (1 to 7) and 8 negative numbers (-1 to -8), hence a total of 16 numbers represented.

The statement is correct as 2's complement allows an extra number.

Question 32 (c) (i)**Sample answer:**

$$G = (A + B) \cdot \overline{(A \cdot B)}$$

Question 32 (c) (ii)**Sample answer:**

Question 32 (c) (iii)**Sample answer:**

The globe will start unlit.

When A is switched on, the globe will light.

It will go out when B is switched on.

It will light up again when A is switched off.

Question 32 (d)**Sample answer:**

$$26.125_{10} = 11010.001_2$$
$$= 1.1010001 \times 2^4$$

$$\text{Sign bit} = 0$$

$$\text{Exponent} = 127 + 4 = 131$$
$$= 10000011$$

$$\text{Mantissa} = 1010001$$

0	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

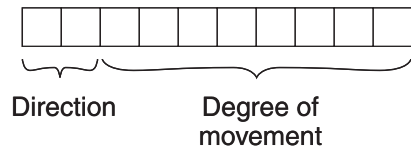
Sign bit Exponent
 8-bit

Mantissa
23 bits

Question 32 (e) (i)**Sample answer:**

The header and the trailer are needed to signal the start and the end of each instruction.

The header and the trailer can also contain things like data transmission error checking bits.

Question 32 (e) (ii)**Sample answer:**

The first 2 bits array represents the directions of:

Stop	-	00
Forward	-	11
Turn left	-	10
Turn right	-	01

The last 8 bits can represent the binary degree of movement.

00000000	-	no turning (either stop or forward)
00000001	-	turning 1°
00000010	-	turning 2°
.		
.		
.		
10110100	-	turning 180°