Software Development Techniques

December 2015

Sample Exam Marking Scheme

This marking scheme has been prepared as a **guide only** to markers. This is not a set of model answers, or the exclusive answers to the questions, and there will frequently be alternative responses which will provide a valid answer. Markers are advised that, unless a question specifies that an answer be provided in a particular form, then an answer that is correct (factually or in practical terms) **must** be given the available marks.

If there is doubt as to the correctness of an answer, the relevant NCC Education materials should be the first authority.

Throughout the marking, please credit any valid alternative point.

Where markers award half marks in any part of a question, they should ensure that the total mark recorded for the question is rounded up to a whole mark.

Answer ALL questions

Marks

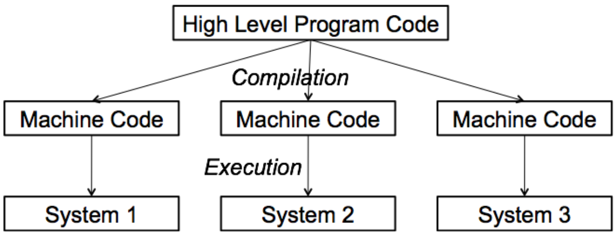
Question 1

a) Some programs written in high level programming languages are compiled. Briefly explain what is meant by *compilation*. **4**

Compilation is the process of taking a human readable (1 mark) representation of code (1 mark), and converting (1 mark) it into a form the computer can understand (1 mark).

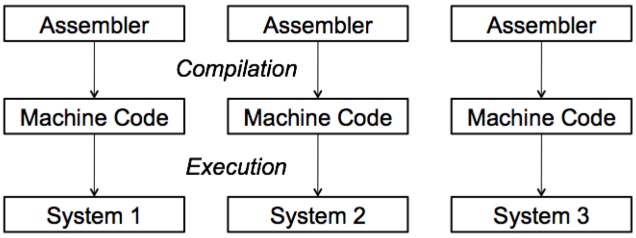
b) Draw ONE (1) diagram that shows the compilation of a program written in a high level language. **3**

The maximum number of marks awarded to this question is 3. Award 1 mark for showing a single high level language. Award 2 marks for showing multiple machine code/systems.



c) Draw ONE (1) diagram that shows the compilation of a program written in assembly language **and** outline how it differs to the compilation of a program written in a high-level language. **3**

The maximum number of marks awarded to this question is 3. Award 1 mark for showing multiple assembler files. Award 1 mark for multiple machine/code systems. Award 1 mark for a brief description that compares each diagram.



Total: 10 Marks

Marks

Question 2

a) Define the term *algorithm* **and** briefly explain what is meant by inputs and outputs. **4**

An Algorithm is a series of instructions (1 mark) that manipulate data (1 mark). Input data is read into the algorithm (1 mark) and the algorithm produces some output depending on its purpose (1 mark).

b) Identify TWO (2) properties of a good algorithm **and** provide a short explanation for each property. **4**

Award 1 mark for stating each property and 1 mark for a corresponding reason up to a maximum of 4 marks. Properties include: Complete; Robust; Efficient; Readable; Maintainable; and Documented.

c) State TWO (2) reasons why we would use pseudocode to write an algorithm. **2**

Award 1 mark for each bullet point up to a maximum of 2 marks:

- ***It means we are not tied to a specific programming language.***
- ***It means we can focus on the logic of an algorithm rather than the language specific features.***
- ***It means we can formally check the correctness on paper.***

Total: 10 Marks

Marks

Question 3

a) The following pseudocode algorithm calculates the area of a floor so that a carpet can be fitted that is the correct size. **10**

Perform a desk-check on this algorithm using a table to show the values in each variable after the execution of each line. You should assume an input value of 50 for length and 50 for width.

```
1.  data width as whole number
2.  data length as whole number
3.  data area as whole number
4.  output “Welcome to the carpet area calculator”
5.  output “What is the width of your floor”
6.  input width
7.  output “What is the length of your floor”
8.  input length
9.  area = length * width
10. output “For your floor you will need a carpet that is:”
11. output area
12. output “Thankyou for using this program.”
```

The maximum number of marks awarded to this question is 10. Award 1 mark in total for correctly stating lines 1-3. Award 1 mark for each correct line from lines 4-12.

Line	*width*	*length*	*area*	*notes*
1	*0*	*0*	*0*	*width declared*
2	*0*	*0*	*0*	*length declared*
3	*0*	*0*	*0*	*area declared*
4	*0*	*0*	*0*	*Output*
5	*0*	*0*	*0*	*Output*
6	*50*	*0*	*0*	*User inputs 50*
7	*50*	*0*	*0*	*Output*
8	*50*	*50*	*0*	*User inputs 50*
9	*50*	*50*	*250*	
10	*50*	*50*	*250*	*Output*
11	*50*	*50*	*250*	*Output*
12	*50*	*50*	*250*	*Output*

Total: 10 Marks

Marks

Question 4

a) State TWO (2) questions that should be considered when choosing a data type. **2**

Award 1 mark for each bullet point up to a maximum of 2 marks:

- ***What kind of information you need to store?***
- ***What kind of manipulations you are going to do to the data?***
- ***What kind of format will be used for output?***
- ***How often you might need to change the representation?***

b) State what data types should be used for the following variables and explain your answer:

i) The name of a student **2**

Award 1 mark for a correct choice and 1 mark for a valid explanation:

String because it is made up of characters

ii) The phone number of a student **2**

Award 1 mark for a correct choice and 1 mark for a valid explanation:

String because it will have a leading zero

iii) The age of a student **2**

Award 1 mark for a correct choice and 1 mark for a valid explanation:

Integer because we usually measure age in whole years

iv) The height of a student in metres **2**

Award 1 mark for a correct choice and 1 mark for a valid explanation:

Real/float because we will have fractional part of a metre

Total: 10 Marks

Marks

Question 5

a) Briefly explain the difference between a bounded loop and an unbounded loop **and** provide ONE (1) example of when we might use each type of loop. **4**

Bounded Loop –When we know how many times we are going to repeat (1 mark). Award 1 mark for relevant example that is clearly suited to a bounded loop.

Unbounded Loop –When we do not know how many times we are going to repeat (1 mark). Award 1 mark for relevant example that is clearly suited to an bounded loop.

Note: A range of valid examples should be credited.

b) Write a short pseudocode program that will print out the multiplication table up to 10 for a given input value. For example, if I used the input value of 3 my output would be: **6**

```
1 x 3 = 3
2 x 3 = 6
3 x 3 = 9
4 x 3 = 12
5 x 3 = 15
6 x 3 = 18
7 x 3 = 21
8 x 3 = 24
9 x 3 = 27
10 x 3 = 30
```

The maximum number of marks awarded to this question is 6. The code should be something similar to this:

```
data userValue as whole number
data counter as integer
counter = 1
input userValue
loop until counter is equal to 10
   result = userValue * counter
   output counter + " x " + userValue + " = " + result
   increment counter
next loop
output "Goodbye!"
```

Marks should be allocated as follows: 1 mark for using the loop; 1 mark for using a counter; 1 mark for ending at 10; 2 marks for choosing a bound loop; and 1 mark for clarity. The pseudocode does not need to be exact but it does need to be clear.

Total: 10 Marks

Marks

Question 6

a) Write a pseudocode algorithm that determines if a given input value is an odd number. It should make appropriate use of variable names and data types. **6**

The maximum number of marks allocated to this question is 6. An example solution is:

```
data userNumber as whole number
data remainder as whole number

output "Please enter a whole number"
input userNumber
remainder = userNumber % 2
if remainder is != 0 then
    output "That's an odd number!"
end if
output "Goodbye!"
```

Marks are allocated as follows: 2 marks for appropriate use of if statement; 2 marks for correct logical condition; 1 mark for appropriate variable names and 1 mark for appropriate data types.

b) Using a nested if statement, write pseudocode that implements the following rule: **4**

If it is raining, wear a “raincoat” and if it is raining and not windy take an umbrella.

You should use the following variables: (1) data coat as String; (2) data umbrella as Boolean; (3) data raining as Boolean; and (4) data windy as Boolean;

The maximum number of marks allocated to this question is 4. An example solution is:

```
If raining == true then
   coat = “raincoat”
   If windy == true then
       umbrella = True
   End if
End if
```

Award up to 2 marks for correct logic and award up to 2 marks for correct nested if structure. For full marks, structure is important, but not syntax.

Total; 10 Marks

Marks

Question 7

a) Construct a truth table for the logical equation A OR B AND C. **8**

A	**B**	**C**	**A OR B**	**(A OR B) AND C**
False	False	False		
False	False	True		
False	True	False		
False	True	True		
True	False	False		
True	False	True		
True	True	False		
True	True	True		

Award 1 mark for each correct line up to a maximum of 8 marks.

A	***B***	***C***	***A OR B***	***(A OR B) AND C***
False	***False***	***False***	***False***	***False***
False	***False***	***True***	***False***	***False***
False	***True***	***False***	***True***	***False***
False	***True***	***True***	***True***	***True***
True	***False***	***False***	***True***	***False***
True	***False***	***True***	***True***	***True***
True	***True***	***False***	***False***	***False***
True	***True***	***True***	***True***	***True***

b) Explain the effect of brackets on the solution to the equation A OR (B AND C). You do **not** need to create a truth table for this question, although it may help your explanation. **2**
Without brackets the logic will be solved in pairs from left to right. By using brackets we are forcing certain parts of the equation to be solved first (1 mark). Therefore in this case (B AND C) will be solved first, and then A OR (B AND C) will be solved (1 mark)

Total: 10 Marks

Marks

Question 8

a) State FOUR (4) advantages of using functions in programming. **4**

Award 1 mark for each bullet point up to a maximum of 4 marks

- ***Breaks problems down into manageable parts***
- ***Simplifies checking and testing***
- ***Make it easier to change/expand programs***
- ***Easier to follow programs***

b) The following pseudocode program is used to display the times table of the user's choice: **6**

```
data userChoice as whole number
output "Please enter the times table you would like to calculate:"
input userChoice
call timesTable(userChoice)
output "Goodbye!"
```

The program uses a function called "timesTable()" to create and display the times table specified in an argument. Write the contents of the function timesTable() in pseudocode.

The maximum number of marks awarded to this question is 6. Answer should be similar to below although syntax does not need to be precise.

```
Function timesTable(needs num as whole number) returning nothing
data counter as integer
counter = 1
loop until counter is equal to 10
   result = num * counter
   output counter + " x " + num + " = " + result
   increment counter
next loop
```

Marks should be allocated as follows: 1 mark for function keyword; 1 mark for having correct contents; 2 marks for argument; and 1 for correct use of it; and 1 mark for returning nothing.

Total 10 Marks

Marks

Question 9

a) Explain the difference between Black box testing and White Box testing. **5**

Black box testing is functional testing (1 mark) and checks the output of the program (1 mark) based on a given input (1 mark). White box testing tests the paths through the code to make sure each is functioning correctly. (1 mark.) It should ensure that all paths through the code are executed (1 mark).

2

b) Briefly discuss why we should perform both black box testing and white box testing.
We carry out both because if we get an error in our data we do not know where it has occurred (1), and even if our data provides the correct result, we cannot be sure that all paths will give the correct result (1).

c) The following code tests the value in the variable i to ensure that it is greater than or equal to zero: **3**

```
Input i
If (i >= 0) Then
   output studentGrade[i]
Else
   output “value out of range”
End if
```

State THREE (3) values would you use as input if you had to perform a simple boundary test on i.

Award 1 mark for each correct value up to a maximum of 3 marks:

i=-1; i=0; i=1

Total: 10 Marks

Marks

Question 10

a) The following class is used to describe objects of type FoodItem. These objects would appear in a menu application for a restaurant. The class is incomplete as it does not contain accessor functions. Write pseudocode accessor functions for this class. **8**

```
Class FoodItem
        data name as String
        data price as whole number

        // Accessor functions to be written here

End Class
```

The maximum number of marks awarded to this question is 8. Award 2 marks per function – syntax can be approximate but must refer to correct member variables (1 mark) and getters must return correct values (1 mark).

```
function setName (needs n as String)
        name = n
end function
function setPrice (needs p as Whole number)
        price = p
end function
function getName() returns String
        return name
end function
function getPrice() returns whole number
        return price
end function
```

b) Briefly explain why we use Accessor functions in classes. **2**

It improves the maintainability of our objects (1 mark). If we want to change the way a name is represented, it can be achieved as all changes have to be done through our accessor functions (1 mark).

Total: 10 Marks

End of Examination Paper

Learning Outcomes matrix

Question	**Learning Outcomes assessed**	**Marker can differentiate between varying levels of achievement**
1	1	Yes
2	2	Yes
3	2, 3	Yes
4	3	Yes
5	2, 3	Yes
6	2, 3	Yes
7	5	Yes
8	7	Yes
9	4, 6	Yes
10	7	Yes

Grade descriptors

Learning Outcome	Pass	Merit	Distinction
Identify and explain the key stages of software development lifecycles	Provide adequate ability to explain the subject matter	Provide detailed and coherent explanation of the subject matter	Provide comprehensive, lucid explanation of the subject matter
Express, design and evaluate algorithms	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Identify and use programming language constructs	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Identify and use common data structures	Demonstrate ability to perform the task	Demonstrate ability to perform the task consistently well	Demonstrate ability to perform the task to the highest standard
Explain and use common algorithms	Demonstrate adequate ability to explain the subject matter; Demonstrate adequate and appropriate use	Demonstrate detailed and coherent explanation of the subject matter; Demonstrate appropriate and effective use	Demonstrate comprehensive, lucid explanation of the subject matter; Demonstrate highly appropriate and effective use
Explain and use test strategies	Demonstrate adequate ability to explain the subject matter; Demonstrate adequate and appropriate use	Demonstrate detailed and coherent explanation of the subject matter; Demonstrate appropriate and effective use	Demonstrate comprehensive, lucid explanation of the subject matter; Demonstrate highly appropriate and effective use
Explain how software is modularised	Provide adequate ability to explain the subject matter	Provide detailed and coherent explanation of the subject matter	Provide comprehensive, lucid explanation of the subject matter