

NSC 2016 CHIEF MARKER'S REPORT

| | | | |
|-----------------------------|------------------------|------------------|---------|
| SUBJECT | INFORMATION TECHNOLOGY | | |
| PAPER | 1 PRACTICAL | | |
| DATE OF EXAMINATION: | 20 OCTOBER 2016 | DURATION: | 3 HOURS |

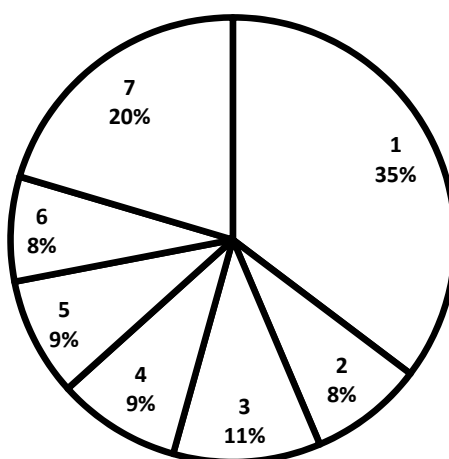
SECTION 1: (General overview of Learner Performance in the question paper as a whole)

Information Technology is classified as a small subject (289 learners). Thirty one centres had learners registered for this examination. It is very disheartening when analysing the results, as there are a few centres that had an average of less than 30%. This indicates that there was no knowledge of Grade 10, Grade 11 and Grade 12 work. This question paper was set according to the correct weighting of cognitive levels, which should have resulted in all learners achieving a pass.

A contributing factor to these poor results is the training of teachers. It should be on-going training and not just once-off. These centres need more support to get their work on the right standard and to ensure that the pass rate increases.

The chart below displays the percentage of learners who achieved levels 1 to 7 in all the results of this examination.

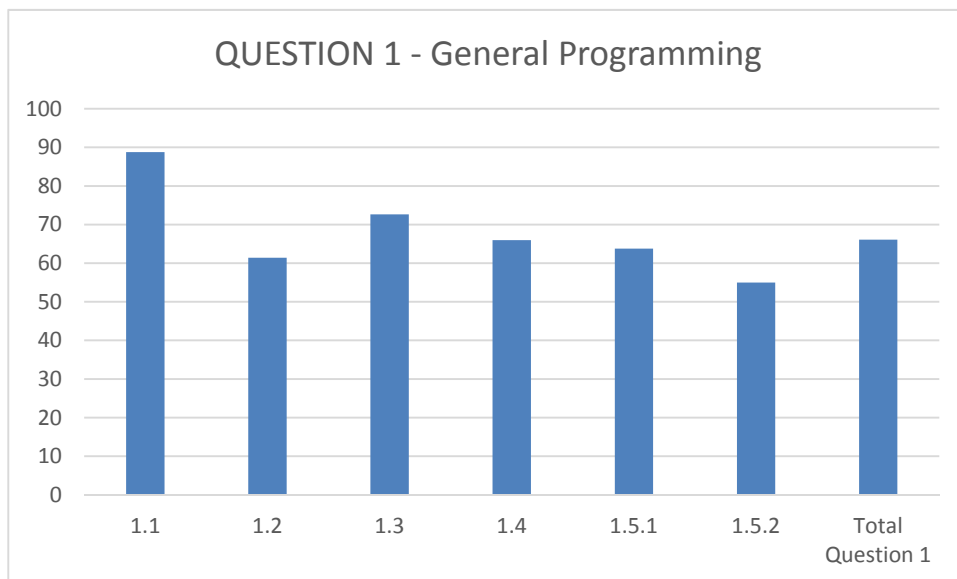
LEVEL DISTRIBUTION FOR THE PROVINCE



SECTION 2: Comment on candidates' performance in individual questions
(It is expected that a comment will be provided for each question on a separate sheet).

QUESTION 1

The chart below displays the average percentage of all marks obtained for this question:



- (a) General comment on the performance of learners in the specific question.
Was the question well answered or poorly answered?

| Average mark from the sample of 100 : | | 61% |
|--|--|-------------------------------------|
| SUB-QUESTION | TOPIC OR ASPECT TESTED | AVERAGE % FROM SAMPLE of 100 |
| 1.1 | Input/output components, number conversion, mathematical calculation, string handling | 77% |
| 1.2 | Mathematical calculation and problem solving, decision making (conditional statements) | 58% |
| 1.3 | Built-in Delphi functions and procedures, mathematical calculation, input/output components and string handling. | 64% |
| 1.4 | Input/output components, string handling, Iteration, conditional statements, mathematical calculation | 61% |
| 1.5.1 | Built-in Delphi functions and procedures, string handling, input/output, decision making, properties of Delphi components | 59% |
| 1.5.2 | Dialog boxes, system date, string handling, string functions and procedures, output components, Delphi built-in functions and procedures | 52% |

(b) Why was the question poorly answered? Also provide specific examples, indicate common errors committed by learners in this question, and any misconceptions.

Most learners managed to complete this question, but the difficulties and where marks were lost are as follows:

1.1 Most learners managed to finish this question and get full marks. A few confused FloatToStr with StrtoFloat and incorrectly used StrtoInt or left it out completely.

1.2 The use of nested IF statements was not easy for some learners.

Many learners were not fully able to translate the more complex mathematical problem of this question into programming code. Many learners used <500 instead of <= 500, for example.

1.3 Many learners were not able to use MOD and DIV effectively. They either swop the two or just don't know about these functions. Many learners used complex solutions which would have wasted their time and, in some cases, lost marks. Mod and Div would have been preferable. The lack of knowledge of how to use DIV and MOD was very prevalent in this question.

1.4 Many learners were not reading the question properly. e.g. When needing to show that the account is paid up, they omitted this instruction and lost 2 marks. The use of a while loop in this question is preferable, although in this case Repeat can also be used. Many learners did not know that they need to use a counter variable in this question and many learners that did use a counter, forgot to initialise it before the Loop instruction. Converting a real number to currency format for display was omitted by many learners. There were many scripts where learners did not deduct the yearly income correctly and the 10% was calculated but not added to the income.

1.5.1 Most learners don't know how to use the random function, or RandomRange function. A common mistake was using Random(6) instead of Random(6) + 1. The definition of the word 'consecutive' was a challenge for some learners.

1.5.2 Many learners did not know how to create a dialog box in code, or were challenged by the word 'dialog box'. There were many learners who did not know how to compile a system date. The use of 'datetostr' was unknown for many learners.

(c) Provide suggestions for improvement in relation to Teaching and Learning

- Teachers must spend more time with algorithms in Grade 10 before starting the Delphi curriculum. Flow charts/pseudocode will enhance learners' problem solving skills and improve understanding of the logic behind more complex conditional programming, i.e. nested IF statements.
- The built-in Delphi functions and procedures must be learnt and taught repetitively so that learners can be confident when using them to solve problems, e.g. DIV and MOD.
- Teachers must stress that learners read correctly and thoroughly.
- Teachers need to explain exactly when to use a while loop and when to use a repeat loop and when to use a for loop.
- Dynamic input methods like INPUTBOX must be explained constantly and memorized by learners.
- Teachers must revise the use of the random function in Grade 12.
- Teachers need to inform learners to not do more than what is required in a question as this wastes valuable time.

(d) Describe any other specific observations relating to responses of learners

The majority of learners knew their Grade 11 syllabus with respect to this question. More complex problem solving is a challenge for some of the learners.

System date was unknown to some learners

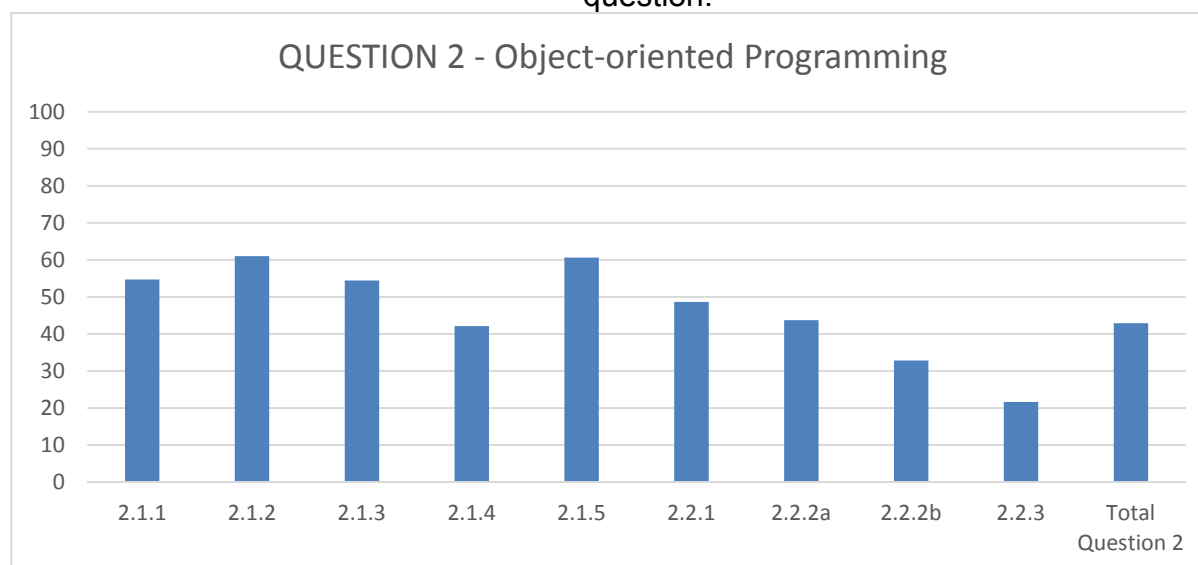
Learners must use 'date' or 'now', and if using 'today' then they must add 'dateutils' to the 'uses' clause.

(e) Any other comments useful to teachers, subject advisors, teacher development etc.

It is important to entrench the basic concepts of the Grade 11 syllabus and make sure the learners can confidently solve problems before moving on to arrays, databases and text files.

QUESTION 2

The chart below displays the average percentage of all marks obtained for this question:



(a) General comment on the performance of learners in the specific question. Was the question well answered or poorly answered?

| Average mark from the sample of 100 : | | 45% |
|---------------------------------------|---|------------------------------|
| SUB-QUESTION | TOPIC OR ASPECT TESTED | AVERAGE % FROM SAMPLE of 100 |
| 2.1.1 | Mutator methods in OOP, Parameter passing | 53% |
| 2.1.2 | Accessor methods in OOP, Conditional statements, mathematical calculations, problem solving | 56% |
| 2.1.3 | Accessor methods in OOP, mathematical calculations, problem solving, conditional statements | 54% |
| 2.1.4 | Accessor methods in OOP, parameter passing, mathematical calculations, | 48% |

| | | |
|----------|---|-----|
| | problem solving, conditional statements | |
| 2.1.5 | Auxiliary and Accessor methods in OOP, string handling | 58% |
| 2.2.1 | Creating an object in OOP, input /output components, conditional statements. String handling. | 49% |
| 2.2.2(a) | User defined methods, specifically creating a Function, parameter passing, Reading from a text file, input/output components, While loops, string handling functions and procedures, conditional statements, mathematical calculations – computing the total of a range of numbers. | 45% |
| 2.2.2(b) | Using a method that was created previously. Using methods of a class previously defined in OOP. Distinguishing between class methods in OOP and user defined methods of the main program. Parameter passing, Conditional statements, input/output components – List box, string handling, repetition (loops, specifically the FOR loop). Properties of Delphi objects | 38% |
| 2.2.3 | Conditional statements, properties of Delphi objects, using methods of a class in OOP, input/output components - combobox. | 28% |

(b) Why was the question poorly answered? Also provide specific examples, indicate common errors committed by learners in this question, and any misconceptions.

2.1.1 Many learners forgot to include a parameter and some learners who had the correct parameter forgot to assign it to the private field.
There were learners who did not know that a procedure was required for this question.

2.1.2 Many learners used a Boolean function instead of a string function. This question was to test that the learners read the question properly because it would be easy to assume that a Boolean function is required. Many learners used the wrong data type to return the value of the function.

2.1.3 The problem for this question is the same as for 2.1.2, only that this is a Boolean function and the return statement has to match that data type. Many learners returned the incorrect data type. The integer parameter was also missing in most answers. Many learners did not add the 'daytotal' to the 'fgroupsize' field.

2.1.4 Many learners forgot to include the two real parameters to this real data type

function and most did not subtract the 'number free' before calculating the cost. Easy marks could have been obtained for checking if a tour guide is required, and then adding that cost, but most learners either did not read the question properly or did not know how to access the private data field, fgroupsize, of the class.

Many learners used Round after dividing by 10 instead of using Trunc.

2.1.5 Most learners forgot to call the RequireTourGuide method when compiling the string. #10 or #13 was required in this question and some learners incorrectly used #9 instead.

2.2.1 Many learners did not know how to use a ListBox, i.e. extract an item from the ListBox. Most learners do not understand or know how to handle the implementation of a class. Many learners did not know how to instantiate an object (They did not use the constructor Create correctly). They used the constructor in the same way as using a procedure. Many learners created a Procedure when they were supposed to create a Function. Many learners did not know how to input data from a Check Box. Many learners used MessageDlg instead of Showmessage. The latter is preferable.

2.2.2.(a) Many learners forgot to initialize a counter or a variable used in calculating the total of a series of number before entering the loop. (i.e. set to 0 before going through a range of numbers in the calculation). Many learners did not know that a counter needed to be used. Using Pos and Copy and Delete when extracting different contents of a long string was a skill not available to some of the learners.

Many learners forgot to return a result at the end of this DetermineDayTotal function. Many learners wrote it as a procedure and that was possibly because they read the question too fast.

Some learners mistakenly created the DetermineDayTotal function in the Class definition and not in the Main unit.

2.2.2(b) The use of constants in the main program as parameters for the CalcAmount method from the class, was difficult for most learners This is an excellent example of using the methods of a class. It is very intricate and difficult to do if one has not had lots of experience in using class methods, therefore this question tests that skill effectively.

Many learners were not able to implement a class, i.e. they were not able to use the methods defined in the class.

Extracting items from a Listbox was a challenge for many learners because this component is rarely used.

Many learners did not know how to add a string to a ComboBox.

2.2.3 Many learners did not know how to check if there are any items in a Combobox. As with 2.2.2(b), many learners did not know how to use the methods in the class.

Many learners omitted this question.

(c) Provide suggestions for improvement in relation to Teaching and Learning

- Writing methods with or without parameters needs much more focus in the teaching and learning process.
- Learners need to know that they must follow instructions explicitly, i.e. when asked to create methods in a class unit, they must know to use that unit to create the methods and they must know not to write the class methods in the main unit which will use the class.
- It is important to highlight the difference between using a Procedure and using a

Constructor in OOP.

- Teachers must, repetitively, discuss the differences between a Mutator method and an Accessor method.
- Teachers must make use of the word instantiate in teaching because many learners might not know what that word means.
- The concept of OOP is very difficult unless they are introduced to it at grade 10 level in more detail, using the components and object inspector of Delphi and providing an introduction to the concept of re-using code.
- The differences between a function and a procedure and how to use those methods, needs more focus and understanding by the learners.
- Learners need to know that they must ALWAYS check to see if a Text file exists BEFORE using the 'Assignfile' statement.
- Teachers must make sure learners use a While loop to navigate through a Text file.

(d) Describe any other specific observations relating to responses of learners

If a learner has had a wide range of practice using OOP, then he/she would have found this question easy to answer.

One or two learners changed the parameters of the constructor Create. Modifying existing code is not allowed. However, there was no instruction to not do so, therefore no penalty was deducted.

(e) Any other comments useful to teachers, subject advisors, teacher development etc.

The concept of calculating a running total has not been mastered sufficiently by many learners in order for it to be used successfully in more complex problem solving.

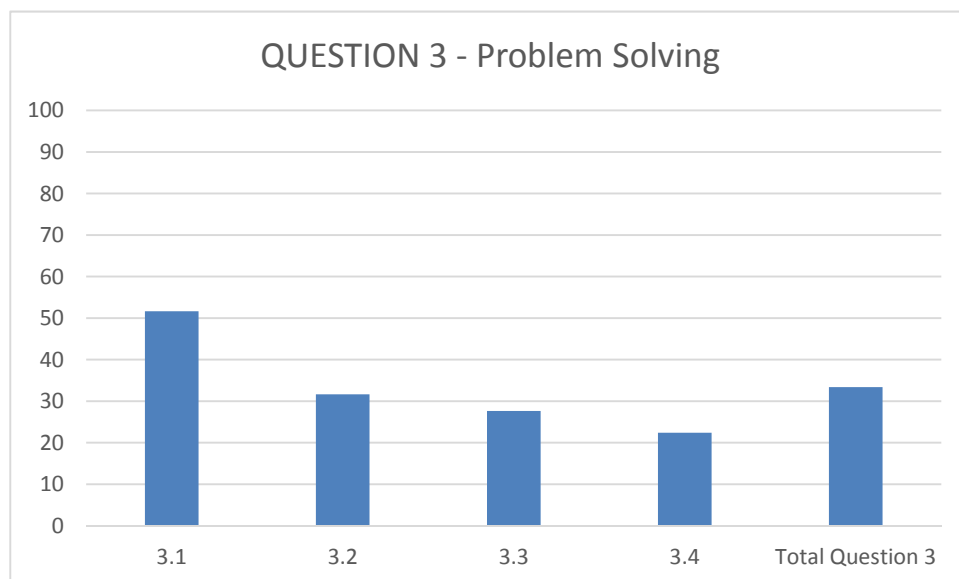
Grade 12 learners need to revise Text file manipulation and string handling.

Learners must be taught to check the parameters of the constructor method.

If a learner does not complete the class as in Question 2.1, but creates the class methods in the main program, then he/she will lose all the marks for the class definition question.

QUESTION 3

The chart below displays the average percentage of all marks obtained for this question:



- (a) General comment on the performance of learners in the specific question. Was the question well answered or poorly answered?

| Average mark from the sample of 100 : 39% | | |
|--|--|------------------------------|
| SUB-QUESTION | TOPIC OR ASPECT TESTED | AVERAGE % FROM SAMPLE of 100 |
| 3.1 | Output components and formatting of output, Iteration (loops), 2D arrays, string handling, decision making (CASE and IF statements) | 54% |
| 3.2 | Output components, 2D arrays, Iteration (loops), logical comparisons, 1D arrays | 37% |
| 3.3 | Input/output components, mathematical calculations, 1D arrays and 2D arrays, logical comparisons, string handling, decision making (IF statements), Iteration. | 34% |
| 3.4 | Input/output components, mathematical calculations, 1D arrays and 2D arrays, logical comparisons, string handling, decision making (IF statements), Iteration. | 31% |

- (b) Why was the question poorly answered? Also provide specific examples, indicate common errors committed by learners in this question, and any misconceptions.

3.1 Displaying a 2D array in a RichEdit should be done with 2 for loops, one for rows and one for columns, but learners were not totally confident with this method and some used a StringGrid component instead, and they were not penalised because there was no instruction to not add components. In future, however, the instruction

will be there to not add components.

Many were not able to produce a data structure to find directions North, South, East, West. They could have used either a case statement or If statements.

Too many output lines are displayed if learners do not build a string for display.

3.2 Navigating through a 2D array and comparing elements with content from a single array was a challenge for most learners. This question also included navigating through a string to compare characters of that string with items in a single array. It was a challenge for most learners.

3.3 This question was difficult in that a string from a 2D array had to be navigated one character at a time to compare with another specific character and to count the equalities. Learners forgot to include a counter variable. Many learners used the wrong character code, i.e. They used the first letter of the activity instead of the given code in the array named Arrcodes.

3.4 This question required deleting a character from a string contained in a 2D array element. Most learners did not achieve this and many did not know how to navigate through a 2D array.

Question 3 required that modular design must be used (ref. Bottom of Page 14 of the question paper) and therefore the mark to call a method to display the array was lost by most learners.

It was assumed that learners would create a method for display of the array including headings since it was used twice in this question, hence modular programming.

The concept of extracting an item from a ComboBox was unknown by many learners.

(c) Provide suggestions for improvement in relation to Teaching and Learning

Learners must be trained in programming examples that include the use of single and 2D arrays in the same problem statement and solution, and to combine these data structures with string handling.

Educators need to highlight the method of building a string within a loop since this concept is not properly understood by most learners.

Learners must be told that they must NEVER add components unless instructed to do so in an examination paper NOR may they change existing code in the data files.

(d) Describe any other specific observations relating to responses of learners

Most learners did not write the display method for Q3.4. Teachers need to take note of this because it is assumed that the learners know that they should have written the method for displaying the array.

(e) Any other comments useful to teachers, subject advisors, teacher development etc.

2D arrays and single array programming is sometimes given less time when teaching, because of more time spent on PAT and databases. Unfortunately, this produces lower marks in the type of challenges that Question 3 presented. Arrays can be taught in a simplified manner so that the more complex problems will be easier to solve.

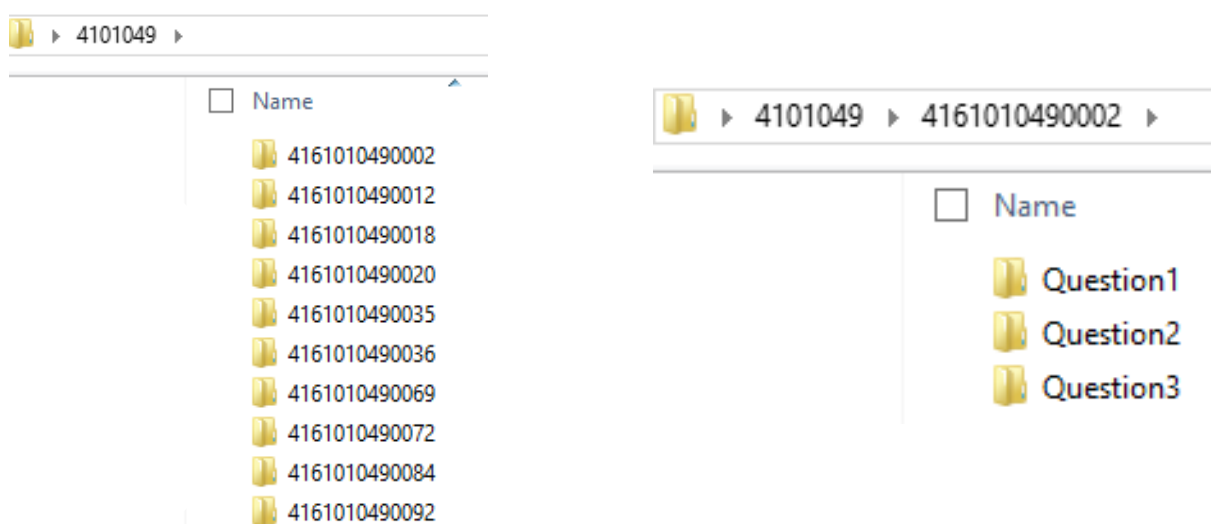
In general, learners need to be reminded about indenting code in order to make the code more readable.

EXTRA INFORMATION

Below are very important instructions regarding the submitting of Practical papers at the examination venue after the examination has been completed by learners.

- The Examination Centre Name (i.e. town name or school name) may not be written or appear anywhere on CD or paper when submitting the learners' answers.
- The numbering of the folders for the learners' exam answers must only contain the examination number of the learner.
- The contents of this examination number folder must contain separate folders for questions 1, 2 and 3. Please take note of the screen shot and instructions on the next page.

Example: The screenshot below on the left represents the contents of the CD for centre number 4101049 which contains 10 learners' exam answers.



The screenshot on the right represents the contents of one learner's answer files.

Printing of exam code:

This is not compulsory but is preferable for efficient and accurate marking.

Printing of code can take place in one of two different ways.

Printing of code may NOT involve making screenshots of learners answers and pasting that into a word document. This is not allowed and marking will then only take place from the CD.

The following instructions describe 2 different methods of printing.

Method 1 for printing (from Delphi):

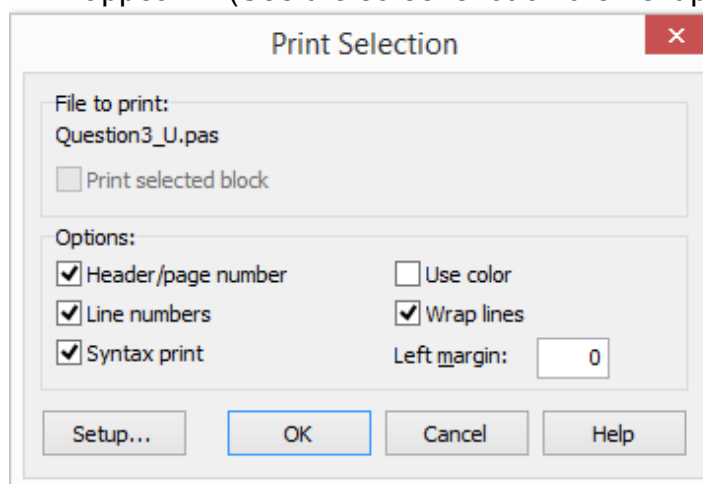


Ikamva eliqaqambileyo!

Make sure the code is visible.

Go to File → Print

The following window will appear: (See the screenshot on the next page)



Make sure to select all the checkboxes as indicated, to ensure that long lines of code are not cut off when printing (wrap lines is essential).

Click on Setup to ensure that printing will be sent to the correct printer.

Click OK.

Method 2 for printing (from Word):

Make sure the Word document is blank before pasting and that there is single line spacing and no spaces allowed before or after paragraphs. Insert automatic page numbering.

Make sure that all learners have typed their examination number as a comment at the top of each program unit, as per the question paper instructions

Open Question 1 program from learner's folder and F12 to get to the code.

Press Ctrl <A> to select all code.

Press Ctrl <C> to copy all code.

Open word document.

Press Ctrl <V> to past into the word document.

Insert a page Break

Open Question 2 and repeat as for Question 1.

Open Question 3 and repeat as for Question 1.

Save the word document with the learner's examination number as the file name and print.
