How do I assess programming?

Programming isn't difficult if you know how to solve problems. If you take the same approach to assessment, it is just as easy. Use the guidance provided below to assess each of the programs provided.

Assessment criteria

The assessment criteria associated with judging a computer program have five simple aspects:

1 Does the program run?

This is not necessarily the most important aspect. A very good program with all the features of readability, understandability, extensibility and general value as a product, might not, at the point of judgement, work. When learning to program (and as an experienced programmer developing a product) much of the time, the program does not work!

However, even though a good program may not run, it should be clear that it can be made to work with only minor modifications.

2 Is the program formatted for readability?

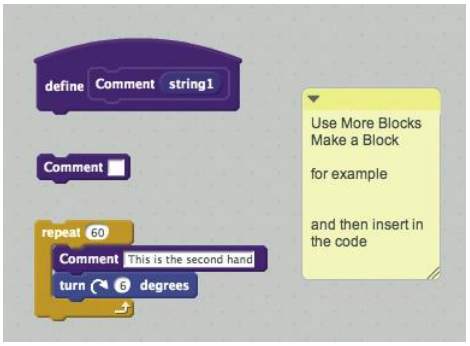
When looking at the code with a squinted eye, does it look formatted with indentations and blank lines to indicate the programming structure of loops, procedures, modules etc?

```
110 REM Nested Loops
120  FOR n1 = 1 TO 10
130   PRINT ""
140   PRINT n1;
150   PRINT " men went to mow, went to mow a meadow"
160    FOR n2 = n1 TO 1 STEP-1
170     PRINT n2;
180     PRINT " men"
190    NEXT n2
200   PRINT "and his dog, went to mow a meadow"
210  NEXT n1
220 END
```

3 Is the program 'commented' for understandability?

Within the code are there comments that indicate what the program is doing at that particular point? Are the contents of variables described at the point when they are first used or changed? Is the purpose described at the start of the procedure and when it is called? Are the names of the programs, variables and procedures indicative of their purpose?

In BASIC and Visual BASIC a comment is a REM statement, in Python #, in Java and MySQL// or/* */ ; in Scratch comments can be added or a block created and inserted.



4 Is the program extensible and portable?

Can the code created by the student be used in other contexts? Do they understand how the code could be adapted for other uses, either in the same program or in other applications? Has the code the potential to be used by others in the development of other programs? This is a high level skill which requires the student to be aware of the readability and understandability issues as well as the potential for the code to be ported into other languages/computers.

5 Is the product good?

This aspect of the quality of code is the most generic and overarching. It is about making a judgement about the 'product'; that is, the user experience of the program. Judgements can be made by asking whether it meets the criteria (original specifications for the design or regulations/rules/requirements associated with the product) and the heuristics (the general 'rules of thumb') associated with the specific area of programming. How effective is the program, how well does the program meet the requirements, and how efficient is the program, in terms of coding time, running time and resources?

An important observation: the programming language and the programming environment will determine whether certain aspects can be or must be implemented.

Program 1: 'One man went to mow' in Scratch	
Does the program run?	
Is the program formatted for readability?	
Is the program 'commented' for understandability?	
Is the program extensible and portable?	
Is the product good?	

Program 2: 'One man went to mow' in BASIC	
Does the program run?	
Is the program formatted for readability?	
Is the program 'commented' for understandability?	
Is the program extensible and portable?	
Is the product good?	

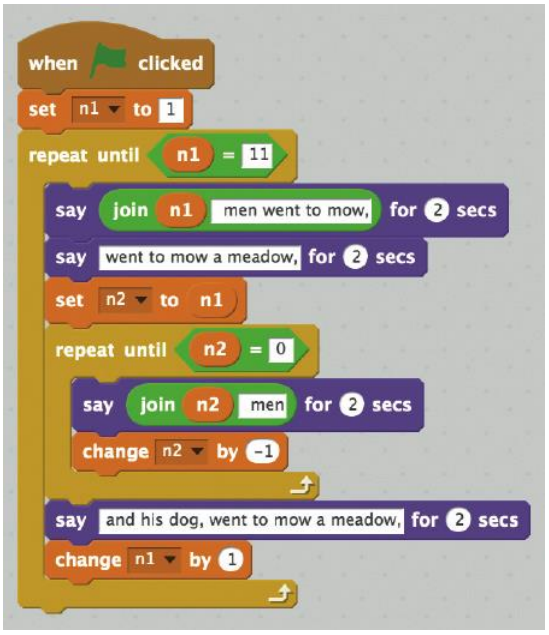
Program 3: 'Sine wave' in BASIC	
Does the program run?	
Is the program formatted for readability?	
Is the program 'commented' for understandability?	
Is the program extensible and portable?	
Is the product good?	

Program 4: 'Calendar' in pseudocode	
Does the program run?	
Is the program formatted for readability?	
Is the program 'commented' for understandability?	
Is the program extensible and portable?	
Is the product good?	

Program 1

Task: Using the online version of Scratch, create a program that prints the grammatically correct words for the song ‘One man went to mow’.

To test this program go to scratch.mit.edu/projects/25828736.



Program 2

Task: In BASIC, create a program including iteration and selection that prints the grammatically correct words for the song ‘One man went to mow’.

```
110 REM Nested Loops
120  FOR man = 1 TO 10
130   PRINT “”
140   PRINT man;
150    IF man = 1 THEN PRINT “ man”;
160    IF man > 1 THEN PRINT “ men”;
170   PRINT “ went to mow, went to mow a meadow”
180    FOR countdown = man TO 1 STEP-1
190     PRINT countdown;
200      IF countdown = 1 THEN PR3INT “ man”;
210      IF countdown > 1 THEN PRINT “ men”
220    NEXT countdown
230   PRINT “ and his dog, went to mow a meadow”
240  NEXT man
250 END
```

To test the program, copy the code into this emulator:

- www.calormen.com/jsbasic

Program 3

Task: Write a program that creates a graphical impression of a sine wave.

Hint: SIN(x) produces a value between -1 and +1 for values of x between 0 and 2*PI

```
100 REM X counts the number of steps (*s) to be printed
101 REM Y can calculates the height of the *
102 REM X/5 gives 36 * for every cycle
103 REM Z counts the spaces before the * is displayed
104 REM 22/4/13 version 1.04
110 FOR X = 1 TO 1000
120 LET Y = (SIN(X/5)*20)+30
130 FOR Z = 1 TO Y
140 PRINT “ “;
150 NEXT Z
160 PRINT “*”
```

```
170 NEXT X
180 END
```

Program 4

Task: Write, in pseudocode, an algorithm for printing out the dates of the year using two arrays:

- One containing the months of the year and the number of days in each month.

  DATA “January”, 31, “February”, 28, “March”, 31, “April”, 30, “May”, 31, “June”, 30, “July”, 31, “August”, 31, “September”, 30, “October”, 31, “November”, 30, “December”, 31

- One containing the days of the week

  DATA “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”, “Sunday”

Declare the arrays:

M$(12) to store the names of the months (string)
D(12) to store the days in each month (number)
W$(6) to store the days of the week (string) W$(0) is Monday.

Read the data:

REPEAT 12 names and number of days
REPEAT 7 days of the week

Variables:

A counts the months 1 to 12
B counts the days from 1 to the number in the month
C counts the days of the year. Set C to be the day of January 1st.

Nested loops:

from 1 to 12 (January to December)
from 1 to 28, 30 or 31 (depending on month)

output the day of the month, the month and the day of the week w$(C modular 7)

increment C

DATA “January”, 31, “February”, 28, “March”, 31, “April”, 30, “May”, 31, “June”, 30, “July”, 31, “August”, 31, “September”, 30, “October”, 31, “November”, 30, “December”, 31

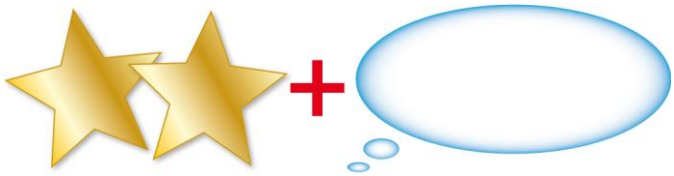
DATA “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”, “Sunday”

Example judgements using the five aspects

Formative assessment has an important impact on learning. The example judgements on the programs provided below are expressed as guidance for students. Did you make similar points when assessing the programs?

Program 1: ‘One man went to mow’ in Scratch

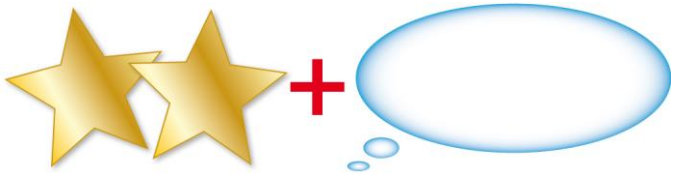
This program runs and produces text on the screen with an appropriate and interesting graphic. The colour coding and shapes of the programming structures in Scratch aids readability and the nested loops are clear. The variables have simple, single letter names and there is no indication of what they represent. The graphic helps indicate the context of the application. There is little scope for application in other contexts. The overall value of the program is limited because of the specification and programming environment. It did not meet the specification because the grammar of the output is incorrect. For example, ‘One *men* went to mow’. The exercise requires a selection statement so that, if the value is 1 then it prints ‘man’ and if it is more than 1 it prints ‘men’.

2*s+wish 

The formatting shows the structure of nested loops and the coding runs. You should consider the use of 'if else' statements to prevent the grammar error of '1 men went to mow'.

Program 2: 'One man went to mow' in BASIC

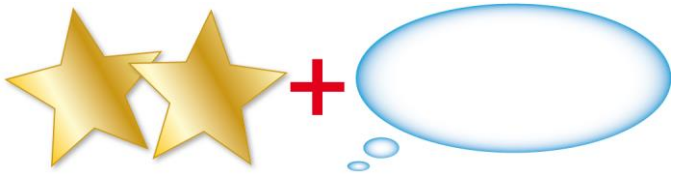
This program runs and produces text on the screen. The code is formatted to aid readability by successive indents for the nested loops. There is little scope to add blank lines but there is only one REM statement to help structure the program. The variables have names that represent their use. There is no description of the process or the context of the application. There is little scope for application in other contexts. The overall value of the program is limited because of the specification and programming environment. It did meet the specification because the grammar of the output is correct, producing 'one man' and 'two men'. The code includes iteration ('for next' loops) and selection ('if then' statements).

2*s+wish 

The formatting shows the structure of nested loops and the coding runs. The output is grammatically correct. You should consider the use of REM (comment) statements to make the working of your code clear and to help others or yourself modify it later.

Program 3: 'Sine wave' in BASIC

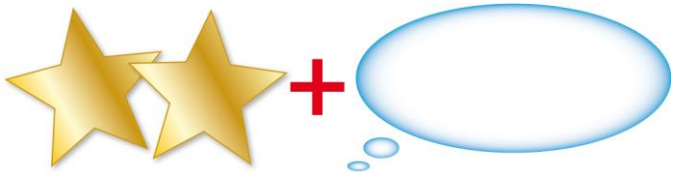
The program runs with no errors. There is no formatting, which suggests the author does not fully understand the significance of nested loops. However, the variables are clearly described and the algorithm indicated through the REM statements. Perhaps the programmer is aware that the application can be used to demonstrate the patterns of cosine and tangent and so there is potential extensibility. The graphic does not run smoothly, because when the sine value is low there are fewer steps than when the sine value is high, making the speed of rendering variable.

2*s+wish 

The coding is efficient, using a few well-defined variables and creating an effective visual that meets the requirements. To ensure that the program is readable, formatting should be applied to the code to indicate the nested loop structure.

Program 4: 'Calendar' in pseudocode

This has the potential to run with no error. Conventional syntax is used to indicate variable and array names and structures. The code is sectioned, systematically dealing with initialisation, processes and data. However, there is no explanation of the algorithm and how the day of the week is calculated. An example of output would be useful in helping understand what the program does.

2*s+wish 

The coding is efficient, using a few well-defined variables and arrays with conventional syntax/names. The code is clearly structured. You should add more comments to make the algorithm and output more explicit.