Examiners' commentary 2018–2019

CO1112 Creative computing I: image, sound and motion – Zone B

General remarks

Overall, performance on this paper was reasonable, with the average mark being a low second class. There were a handful of first class results, none of which were exceptionally strong, and the highest of these being around 80 per cent. There were more fails than first class passes, however most candidates did obtain a pass for this examination.

What follows is a brief discussion of the individual questions on this paper, with hints and explanations of the answers expected by the examiners.

Comments on specific questions

Question 1: Creativity

Only about a third of candidates chose this question, and performance on it was generally very poor. Only a small handful of candidates obtained sufficient marks to scrape a pass for the question, and no candidate obtained more than half of the available 25 marks.

To achieve good marks for part (a), candidates needed to identify and describe three ways in which creativity in visual arts is enhanced through the use of a computer. It was also essential to describe any software or peripherals that might be useful. Examples might include being able to do things such as very fine work, that are otherwise physically difficult; being able to do colour work, sometimes more accurately, possibly with the limitation that screen colour is different from print colour; producing patterns and recursive artefacts; producing mosaic portraits.

For part (b) examiners hoped for answers that showed thought and insight rather than specific technical knowledge, though technical knowledge was awarded marks too. Examples might include computerised choreographers' notation; animation packages to try out dance moves; the use of technology as part of performance; autocueing for stage plays; the use of Internet-driven theatre plots and so on. Another area is that of musical performance, such as improvisation, and other improvisation formats. This is a rich area that could have been answered well simply through thought and insight.

Part (c) was answered reasonably well, though not all candidates fully responded to the second part of the question. For part (c)(i), an algorithm is a step-by-step description of a solution to a problem. Another approach could be that it is a description of how to do something; a recipe might be seen as analgorithm.

For part (c)(ii), some might argue that algorithms have no place in computational creativity: if there is an algorithm then there is no creativity; while others might argue that it is the algorithm that is the creative entity.

Part (d) refers to an exercise in the subject guide, so it was expected that candidates could describe the artwork as well as the impact of cultural context on how it is perceived.

Question 2: Data, sound and motion

This question was chosen by half of the candidates, and performance on it was average, with a couple of good responses.

Part (a) was a straightforward question and required candidates to demonstrate that they understood what both sampling and compression actually are. There was clearly some confusion about the difference, with not all candidates showing understanding of how each is done. At a very minimal level, examiners expected candidates to mention that audio sound is converted to a digital representation via sampling. Sample rate is an issue and relates to the quality of the sound and the size of the resultant file. Files that are already in a digital format can be compressed to save space, but there is the possibility of information loss.

Part (b) is material from the subject guide, and most candidates were able to explain that pitch and volume are controlled by moving the hands towards and away from the device, left and right-hand respectively. An overall description of the device was required too.

Again, most candidates were able to say, for part (c), that the Boolean type is either a variable or a function that can take on the values true or false.

Few candidates gave a good answer for part (d). Examiners were expecting candidates to demonstrate that they knew that `frameRate()` sets the number of frames that are output per second, which most candidates did. However, not many discussed the role it plays in animation, nor constraints and compromises. In relation to animation, this is how many times to 'refresh' per second. Concerns include that too fast a frame rate could overload the processor, while too slow a rate makes animations jumpy.

Question 3: Colour, shape and structure

This was a very popular question chosen by almost all candidates, and with the highest average performance. Although there were some excellent answers, obtaining nearly all of the available 25 marks, there were also many weak responses, obtaining less than 10 marks.

For part (a), most candidates described either the work of Kandinsky, who developed a theory of the different moods and impacts different colours have, and their relation to sounds; or the work of Itten, who developed theory on colour contrasts and how to combine colour. Candidates were also expected to connect their responses to current ideas, which not alldid.

Part (b) required candidates to name and describe RGB and HSB, and to describe the main differences. An example of an appropriate response might be that the former is a colour mixing approach, while the latter uses the colour, the depth or intensity of colour and the luminosity. Most candidates were also able to give examples of when one would choose a particular approach, such as that HSB is more appropriate for printing.

For part (c), descriptions of transparency were weak but the mechanism used in *Processing* – the Alpha channel – was mentioned correctly by almost all candidates.

Responses for part (d) were good, and examiners accepted algebraic – in terms of the matrix manipulations – or visual explanations asequally sufficient. Most candidates also gave the required examples.

Part (e) showed understanding, but sometimes explanations were a little unclear. The operations have the same effect in some contexts if the rotation is around the same axis that the translation is a cross. However, in general they do not, and this was the most important part to explain. Examiners accepted justification that was algebraic, or by examples, or by graphical drawings.

Question 4 : Generative systems

This was answered by almost all candidates, with generally reasonable to good answers.

For part (a), most candidates were able to explain that the initiator is `state` variable with value "F" and the generator is `F_rule` variable with value "F[-F]+F".

Part (b) required both an explanation of what the `noLoop()` function does, as well as why it is appropriate, to obtain full marks. The `draw()` method draws the whole pattern and there is no need to redraw it on each frame.

For part (c), most candidates were able to say that the brackets save and restore the current drawing position, though for full marks they also needed to mention that the orientation is also saved and restored as part of this. In addition, the `pushMatrix()` and `popMatrix()` functions are used to achieve this.

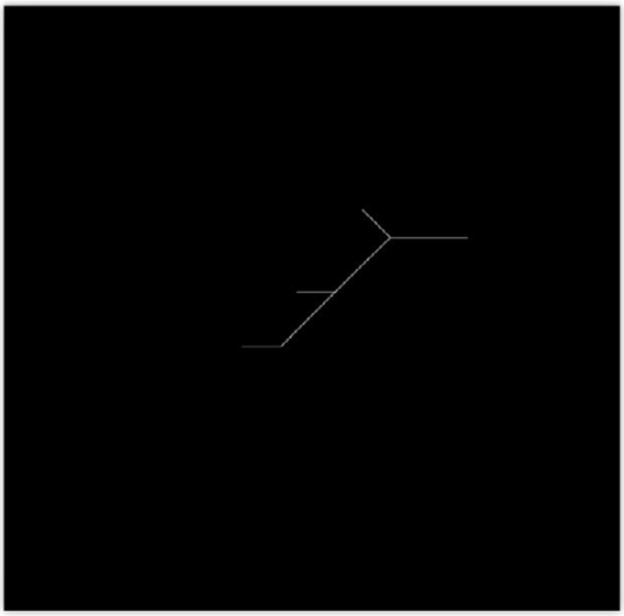
For part (d), almost all candidates correctly identified that the code does not make use of recursion. An example justification would be tomention that recursion would require a call to `substitute` within the `substitute()` method itself, and to clarify thati nstead, itcallsthe `substitute()` method multiple times within a `for` loop (lines 12–13).

Part (e) was answered well, with the response:

```
--FF[+F]-FF[+F][+-FF[+F]]
```

Candidates did not always give complete answers for part (f). The operation ofdrawing a line from the current origin to point (d,0) – or in this to point (50,0) – was usually correctly given. However, some candidates failed to include that it also moves the origin or drawing head to the end of the line, or to point (50,0).

Finally, for part (g), a drawing similar to the one below was expected, with co-ordinates marked at (600,300) for the first point and around (309,441) for the last.



Question 5: Motion and 2D graphics

This was the most popular question, answered by almost all candidates, and with generally good performance on it.

For part (a), the diagram should have shown the size of screen (800x800); background colour (0), or black; ball of diameter (50), centred at (200,200); and filled with colour (0,255,0), or green.

For part (b), a sensible answer would explain that the code checks whether the ball has reached the left or right edges of the screen, and if so, the ball's horizontal velocity is reversed.

For part (c), only some candidates showed understanding that a downwards acceleration, namely gravity, is being applied in the *y* direction, and therefore an upwards travelling ball will eventually come to rest momentarily in the negative *y* direction and then start moving in the positive *y* direction.

Part (d) required candidates to provide one-sentence definitions for the concepts of velocity and acceleration. Despite this being a bookwork question, with appropriate definitions given in the subject guide, not all candidates included direction as being central to these definitions.

For part (e), most candidates were able to explain that there is acceleration in the *y*

direction, implemented on line 22 as a change in vertical velocity.

Part (f) could be simply answered by either deleting line 14 ("background(0);") or moving it to the setup() function, and then adding the following three lines at the start of the draw()function, that is, after line 13:

```
fill(255,50); // or other small alpha value
rect(0,0,width,height);
```

```
fill(0,0,255);
```

Examiners also expected the requested explanation for full marks to be awarded. Finally, for part (g), a simple response of modifying line 20 to read:

```
dy = -0.95*dy;
```

obtained half the available marks. However, this will run into difficulties when *dy* gets smaller than the radius of the ball because adding *dy* to *y* might not be enough to get it above the ground level again. So we would need to add a second test to line 18:

```
if ( y > height - ballSize /2 && dy > 0)
```

Explanations that showed understanding of this obtained the rest of the available marks.

Question 6: 3D motion and sound

This was one of the less popular questions, but it was reasonably answered by those who attempted it.

Part (a) was generally well done, and candidates were able to produce diagrams with a large sphere of radius 80 in the centre of the screen (400,400,0), and shown in wire- frame with green lines. Most candidates also included the small sphere of radius 15, 250 units to the right of it, at point (650,400,0), and also drawn in wire-frame but with redlines.

For part (b), again most candidates correctly answered for part (b) that the planet appears to rotate around its y-axis, though not all explained that this is because you can see the wire-frame lines moving. It takes 8 seconds to complete a full rotation.

For part (c), candidates were able to explain that the command controls the number of vertices used to draw the sphere, or said that it specifies the number of segmentsused per full circle revolution. Not all explained advantages and disadvantages, such as that using a lower number renders faster, but the rendering is less smooth.

Part (d) was not always answered comprehensively. In this instance, the behaviour would change, because the rotation would happen around the left-hand side of the screen, on the y-axis at global (x-pos,0) rather than at global (x-pos,400). In general, the order of transformations matters unless they are all only translations, and examiners were expecting candidates to demonstrate understanding of this.

For part (e), it is possible to use the `camera()` function, which takes 9 parameters: eyeX, eyeY, eyeZ, lookX, lookY, lookZ, upX, upY, upZ. An alternative approach, used appropriately by some candidates, utilised the `beginCamera()` and `endCamera()` functions.

Most candidates showed good understanding of the audio aspects of part (f). To use the sound library, the required command is:

```
import processing.sound.*;
```

A simple approach for part (f)(ii) might be to create a new `SoundFile` global variable at the start of the sketch, for example "`SoundFilemyMusic;`". This would need to be initialised in the setup() method, for example "`myMusic=newSoundFile(this, "space.wav");`". Finally, the play method would need to be called, also within the `setup()` method, for example "`myMusic.play()`".