

THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS
--

UNIVERSITY OF LONDON

CO1112 ZA

BSc, CertHE and Diploma Examination

CREATIVE COMPUTING AND COMBINED DEGREE SCHEME

Creative Computing 1: Image, Sound and Motion

Thursday 17 May 2018: 14.30 – 17.30

Time allowed: 3 hours

There are **SIX** questions on this paper. Candidates should answer **FOUR** questions. All questions carry equal marks, and full marks can be obtained for complete answers to a total of **FOUR** questions. The marks for each part of a question are indicated at the end of the part in [.] brackets.

Only your first **FOUR** answers, in the order that they appear in your answer book, will be marked.

There are 100 marks available on this paper.

A handheld calculator may be used when answering questions on this paper but it must not be pre-programmed or able to display graphics, text or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

© University of London 2018

## Question 1      General

(a) For the following questions, give an answer of **T** (if the statement is true), or **F** (if the statement is false). You need only write down the question number, and your answer of **T** or **F**. Each correct answer is worth 1 mark. NOTE: there is no negative marking in this question; an incorrect answer will simply receive a mark of zero; other marks already obtained will not be affected.

- i. There is no representation for the number zero (0) in the Roman numeral system.
- ii. Brunelleschi worked out the principles of perspective, namely that parallel lines in reality should be depicted as converging to a vanishing point.
- iii. Andy Warhol was involved in the work at the Bauhaus.
- iv. The Sierpinski triangle is an example of a fractal.
- v. The pitch of a sound is related to the amplitude via the transformation of a logarithm.
- vi. Pulse Code Modulation is a type of sampling process.
- vii. The highest frequency component that we can measure for a sampling rate of  $SR$  is  $2 \times SR$ .
- viii. 44100 kHz is a standard sample rate for CD quality audio.
- ix. `pushMatrix()` is used to save the current coordinate system transformation in *Processing*.
- x. The Blind Watchmaker algorithm is an example of a user-guided genetic algorithm.
- xi. An L-system is a string rewriting system, where rules are applied in parallel in any one iteration.

[11]

(b) Name **ONE** woman who was part of the Bauhaus.

[1]

(c) What is the output of the following calculation?

$$\begin{pmatrix} 4 & 3 & 3 \\ 0 & 2 & 5 \\ 6 & 5 & 7 \end{pmatrix} \cdot \begin{pmatrix} 8 & 3 \\ 4 & 1 \\ 2 & 0 \end{pmatrix}$$

[2]

(d) Name **TWO** bitwise logical operators provided by *Processing*.

[1]

- (e) Name **TWO** different audio compression formats. [1]
- (f) Define Rule 30 of a 3-bit 1-D cellular automaton. [3]
- (g) Starting with a single bit set to 1, in the centre, show the output of the above automaton after four generations. [3]
- (h) A biomorph, in the context of Dawkins' visualisation of evolutionary processes, has **THREE** important processes or components associated with it. What are these? [3]

## Question 2 History and Creative Thinking

- (a) Describe how we represent numbers using Roman numerals, up to 2000. What is meant by additive and subtractive representations? Use examples relevant to Roman numerals to illustrate your answers. [5]
- (b) Sketch out an algorithm for the addition of two numbers that are given as Roman numerals. The algorithm you sketch out should work for input numbers up to MM. [8]
- (c) *The Bauhaus attempted to rationalise design and production. The formalisation of these creative ideas lends itself to implementation in computational tools for creativity.*  
Explain what is meant by the above statements, and discuss (with justification and substantiation) whether you agree with them. [12]

## Question 3 Colour

- (a) Name **TWO** Bauhaus members or other artists who developed theories of colour, and briefly describe their theories. [7]
- (b) How is colour represented in *Processing*'s `color` datatype? [4]
- (c) What is the difference between a colour scheme and a colour gradient? You should clarify what each is, and how they are different, using appropriate examples. [6]
- (d) Describe **FOUR** different ways in which it is possible to define colour in *Processing*. For each one, show how the colour red would be coded. [8]

#### Question 4    Motion and Interaction

Study the *Processing* sketch below, then answer the questions that follow.

```
1  int holeX;
2  int ballX,ballY;
3  int ballInc=2;
4  int score=0;
5
6  void setup()
7  {
8      size(600,600);
9      frameRate(30);
10     noStroke();
11     textSize(32);
12     resetPositions();
13 }
14
15 void draw()
16 {
17     background(0,0,255);
18
19     fill(255,255,255);
20     text(score,300,40);
21
22     // draw bar
23     fill(255,0,0);
24     rectMode(CORNER);
25     rect(0,275,600,50);
26
27     // draw hole in bar
28     fill(0,0,255);
29     rectMode(CENTER);
30     rect(holeX,300,50,50);
31
32     // draw ball
33     fill(255,255,255);
34     ellipse(ballX,ballY,30,30);
35
36     ballY += ballInc;
37 }
38
39 void resetPositions()
40 {
41     holeX=300;
42     ballX=(int)random(600);
43     ballY=0;
44 }
```



```

45
46 void keyPressed()
47 {
48   if (key == 'X' || key == 'x')
49   {
50     holeX += 5;
51   }
52   else if (key == 'Z' || key == 'z')
53   {
54     holeX -= 5;
55   }
56 }

```

- (a) When run, the code draws a horizontal bar with a hole in it, and a ball that drops from the top of the screen. Draw a sketch of the output produced by this code in the first frame after it is run. State the colour of the background and any objects drawn, give the top-left and bottom-right coordinates of the horizontal bar, the coordinates of the centre of the hole in the bar, the position of any text drawn, and indicate the range of possible starting positions of the ball.

[7]

- (b) The code implements the `keyPressed()` method (lines 46–56) to handle keyboard interaction. *Processing* also provides an inbuilt variable called `keyPressed`, that can also be used for this purpose. Describe the differences and similarities in how these two approaches are used.

[5]

- (c) The code shown forms the basis of a simple game. The idea is that the user moves the hole in the central bar left or right (using the keyboard) to allow the ball to drop through the hole without hitting the bar. Add code to the `draw()` method to implement collision detection between the ball and the bar. If a collision is detected, your code should call the `resetPositions()` method. You do not need to write out the whole program in your answer; just show any new code you would add, and indicate the line number(s) where you would insert it.

[5]

- (d) Add code to the program such that when the ball drops off the bottom of the screen, the `score` variable is incremented and the `resetPositions()` method is called. You do not need to write out the whole program in your answer; just show any new code you would add, and indicate the line number(s) where you would insert it.

[4]

- (e) The given code prints the score to the screen using the system's default font. Describe how you would modify the code to use a different font. Assume that the data folder of your sketch already contains a suitable font file named `MyFont-32.vlw`. State the code you would add to load and use this font, and say where you would insert these lines (by giving the line numbers in the given code after which you would insert each line).

[4]

## Question 5 Generative systems

- (a) Name **ONE** kind of natural object that L-systems are suitable for modelling. What features of L-systems make them suitable for such modelling? [3]

- (b) Consider an L-system with an alphabet  $\{a, b, c\}$ , an initiator  $a$  and the following three generators:

$$a \rightarrow ac$$

$$b \rightarrow bc$$

$$c \rightarrow b$$

Write down the first **THREE** iterations of substitutions on the initiator in this L-system. [3]

(c)

Symbol	Command	Transformation
F	Draw a line of length 25 and move to its end	?
f	?	translate(25,0);
+	Rotate by $\pi/3$ radians (60 degrees)	?
-	Rotate by $-\pi/3$ radians (-60 degrees)	?
[	Open branch (save turtle state)	?
]	Close branch (restore turtle state)	?

The partially-completed table above shows a possible graphical interpretation for an L-system with alphabet  $\{F, f, +, -, [, ]\}$ .

- Complete the **SIX** missing items in the table (denoted by question marks). Where a transformation is missing, give *Processing* code that will implement the given command. Where a command is missing, provide a description of the given transformation in the context of the L-system interpretation. [6]
- Assuming a window size of 200 by 200, a starting position at (100,200), and a starting rotation of  $-\text{HALF\_PI}$ , draw the output of applying the graphical interpretation shown in the table to the string  $F[-FfF][++F]$ . Annotate your drawing with lengths and angles for each part. [3]



(d) Genetic algorithms are models of biological evolution. They have been used in creative applications as a means of exploring large parameter spaces.

i. How is the concept of *selection* used in a genetic algorithm? [3]

ii. When creating a new generation, multiple children may be produced from a single parent in this case. How, and why, is variety introduced among the children? [3]

iii. Suppose that you are writing a *Processing* sketch to evolve L-system patterns using a genetic algorithm. Describe how you might implement a `mutate` method for this program, that takes a symbol string (which could be of any non-zero length) representing the right-hand side of a generator rule, and returns a mutated version of that string. Assume that the method is of the form: `String mutate(String rule)`. You do not need to write code (although you may if you wish), but describe what the method should do, highlight any problem cases that might arise, and suggest how the method should deal with such cases. [4]

## Question 6 3D drawing, animation and object-oriented programming

Study the *Processing* sketch below, then answer the questions that follow.

```
1 Particle p1;
2
3 void setup()
4 {
5   size(500,500,P3D);
6   p1 = new Particle();
7   camera(0,-500,500,0,0,0,1,0);
8   rectMode(CENTER);
9   noStroke();
10 }
11
12 void draw()
13 {
14   background(255);
15   fill(0,255,0);
16   box(500,1,500);
17   p1.draw();
18   p1.update();
19 }
20
21 class Particle      // definition of Particle class
22 {
23   float px, py, pz; // particle position
24   float vx, vy, vz; // particle velocity
25   float ax, ay, az; // particle acceleration
26
27   Particle()        // constructor
28   {
29     px = py = pz = 0.0;
30     vx = random(-5.0, 5.0);
31     vy = random(-5.0, -10.0);
32     vz = random(-5.0, 5.0);
33     ax = ?;
34     ay = ?;
35     az = ?;
36   }
37
38   void draw()
39   {
40     pushMatrix();
41     fill(255,0,0);
42     translate(px,py,pz);
43     sphere(5);
44     popMatrix();
```

```

45 | }
46 |
47 | void update()
48 | {
49 |     vx += ax; vy += ay; vz += az;
50 |     ?
51 | }
52 | }

```

- (a) The sketch makes use of the `Particle` class, which is defined in lines 21–52. The class stores a particle’s current position, velocity and acceleration. In line 6, a new instance of this class is created. When a new instance of a class is created in this way, *Processing* automatically calls the class constructor to initialise the new instance. The constructor is defined in lines 27–36.

We wish to add a gravitational force to the particle as it moves. Lines 33–35 of the constructor are incomplete. State what code should replace the question marks in these three lines so that the particle is initialised with an acceleration of 0.1 units in the downwards direction (*i.e.* a downwards acceleration of 0.1 spatial units per frame per frame).

[3]

- (b) The `update()` method of the `Particle` class (lines 47–51) is incomplete. What code should replace the question mark on line 50 so that the particle’s position is updated appropriately according to its current state?

[3]

- (c) Line 7 of the sketch calls the `camera()` function. Explain the meaning of the nine arguments passed into the `camera()` function on this line.

[3]

- (d) Using a drawing with appropriate labels, describe the output of the sketch when it is run. Show any objects that are drawn and indicate any movement that happens. Specify positions, dimensions and colours where appropriate.

[6]

- (e) Extend the sketch so that **THREE** independent particles are animated rather than just one. You do not need to write out the whole sketch in your answer, just indicate any lines you would modify, delete or add to the original sketch.

[5]

- (f) By modifying the `update()` method of the `Particle` class (lines 47–51), extend the sketch to implement collision detection and response when a particle hits the ground. If a collision is detected you should reverse the vertical component of the particle's velocity. You do not need to write out the whole sketch in your answer, just indicate any lines you would modify, delete or add to the original sketch.

[5]

**END OF PAPER**