

EXAM COVER SHEET**RMIT University Examinations****EXAMINATION DETAILS:**

Course code/s:	COSC1186/1187
Course name/s:	Interactive 3D Graphics and Animation
Date of exam:	Wed 13th June
Time of exam:	9:15am to 10:30am
Duration of exam:	1:15
Total no. of pages (incl. cover sheet):	3

ALLOWABLE MATERIALS AND INSTRUCTIONS TO CANDIDATES:

1. Write your full name and student number on each examination booklet together with the number of examination books used. Students must not write during reading time.
2. This examination paper contains 3 questions
3. Attempt all parts of all questions. Commence each question on a new page. Carry out the instructions on the front cover of the examination booklet
4. This exam accounts for 30% of the total marks for the course.
5. This exam totals 30 marks.
6. Marks for each question are shown.
7. This exam is closed book.
8. Calculators are not permitted.

Question 1

Give a concise answer to each of the following questions:

- (a) What is the “The Red Book”?
- (b) Give the list of libraries which an OpenGL+GLUT program must be linked against when compiling.
- (c) Draw triangles which allow the values of $\cos(30^\circ)$, $\sin(30^\circ)$ and $\sin(45^\circ)$ to be determined.
- (d) What does framerate mean?
- (e) Normalise the vector $(2, 2, -1)$.
- (f) Give the name of a subdivision modeller.
- (g) Give two possible arguments to `glBegin`
- (h) What type of view volume does (i) a *parallel* and (ii) a *perspective* projection give?
- (i) How does a programmer request a double buffer using `glut`?
- (j) What is *morphing*?

(1/2 mark each = 5 marks)

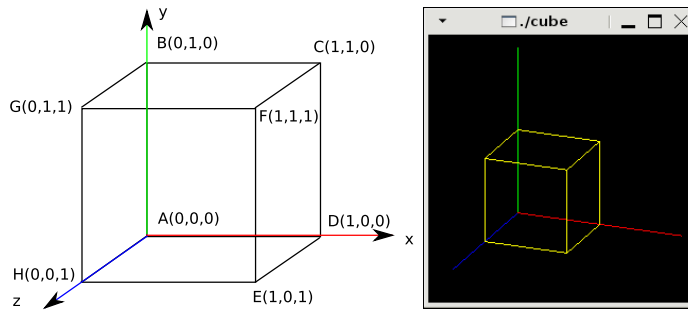
Question 2

A triangle T has vertices $A(4, 1, 1)$, $B(2, 3, 1)$ and $C(1, -2, 2)$. Work out the plane equation of the plane in which T lies.

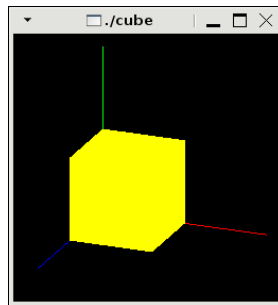
(5 marks)

Question 3

A unit cube is shown below. In the diagram to the left the vertices are labelled and the coordinates given. A screen grab from an OpenGL program which displays the cube rendered as a yellow wireframe is shown on the right.



- (a) Write a code fragment which initialises an array `cubeVertex` with the coordinates of the cube's vertices.
- (b) Write a code fragment which initialises an array `cubeFace` with the indices of the vertices which form each face.
- (c) Write a function `renderCube` which uses OpenGL to render the cube as shown above in the screen grab on the right. Assume viewing and modelling transformations are performed elsewhere, and are not required in this function. The RGB value for yellow is (1.0, 1.0, 0.0).
- (d) A solid rendering of the cube is also required, as shown below:



Write a function `renderSolidCube` to achieve this. Your answer for this question may use your answers to the previous questions, or be given as modifications to them - explain changes where necessary.

- (e) Write the `obj` file for the cube model - just the vertices and faces, no normals or materials.

(4+4+4+4+4 = 20 marks)

THE END