S228/211C, S282/211C

DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science
DT282 BSc. (Honours) Degree in Computer Science (International)

Year 2

SUMMER EXAMINATIONS 2016-2017

OBJECT ORIENTED PROGRAMMING [CMPU2016]

DR BRYAN DUGGAN

DR. DEIRDRE LILLIS

MR. KEVIN. FOLEY

THURSDAY 11TH MAY 9.30 A.M. - 12.30 P.M.

THREE HOURS

INSTRUCTIONS TO CANDIDATES

ANSWER **FOUR** QUESTIONS OUT OF **SIX**. ALL QUESTIONS CARRY EQUAL MARKS.

Question 1

(a) Explain how to set colours in Processing.

(3 marks)

(b) Figure 1 shows an extract from a TAB separated file of expense claims by various politicians

```
Adams, Gerry SF Louth   48643.52
Bannon, James   FG  Longford-Westmeath  40542.72
Barrett, Sean   CC  Dun Laoghaire   13111.75
Barry, Tom  FG  Cork East   50540.23
Boyd Barrett, Richard   PBP Dun Laoghaire   31865.51
Breen, Pat  FG  Clare   51179.23
Broughan, Tommy Lab Dublin North-East   22821.43
...
...
```

Figure 1

Write Processing code for a class called `Expense` that encapsulates the data from a single row of the file. In your solution, include a constructor that takes a parameter of a single line from the file and parses it to assign values to the fields in the class.

(8 marks)

(c) Write code to declare an ArrayList of Expense types.

(2 marks)

(d) Write code for a method called void graphExpenses(). This method should draw a bar chart of the data held in the array list. The barchart should fill the width and height of the screen.

(12 marks)

Question 2

Figure 2 shows a screenshot from a Processing sketch that implements a 2D two player space shooter game similar to *SpaceWar*. Figure 3 shows an extract from the superclass of the classes used to encapsulate the behaviour of the objects in the sketch.

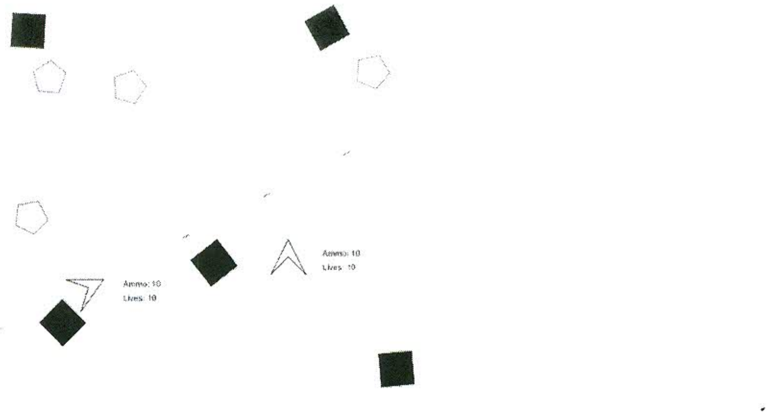

Figure 2

```
class GameEntity
{

  PVector position;
  PVector look;
  PVector velocity;

  float scaleF = 1.0f;
  float speed = 100.0f;
  float timeDelta = 1.0f / 60.0f;
  boolean alive;

  GameEntity()
  {
    position = new PVector();
    look = new PVector(0, -1);
    theta = 0.0f;
    alive = true;
  }


  void update()
  {
  }

  void draw()
  {
  }

}
```

Figure 3

(a) Write code for a subclass of `GameEntity` that encapsulates the behaviour of the player controlled ship from Figure 3 that:

  i. Moves in the direction it is heading in response to the W and S keys (3 marks)

ii. Rotates left and right in response to the A and D keys (4 marks)

iii. Is drawn at the correct position and rotation (controlled by the fields `pos` and `rotation`).

(8 marks)

(b) The ship in Figure 2 can fire bullets at a rate of 5 bullets per second. Explain in detail how you would achieve this. Assume that bullets get removed from the scene when they go outside the bounds of the screen.

(10 marks)

Question 3

(a) What are *access modifiers* in Java used for? Include examples in your solution.

(6 marks)

(b) What is meant by the keyword `static` when applied to a field in Java?

(2 marks)

(c) In git, what does the term "merge conflict" mean? How is a merge conflict resolved?

(10 marks)

(d) Compare the Java `ArrayList` and `Hashmap` collection classes. Be sure to discuss the efficiency of each collection and include a short example of each in your solution.

(7 marks)

Question 4

(a) What is the minimum number of edit operations required to make the string “BigLez “ into the string “Sassi”? Construct a Levenshtein distance matrix to solve this.

(7 marks)

(b) Figure 4 presents an extract from a class that encapsulates a dynamic 2D matrix of floats in Java.

```
package ie.dit;

public class Matrix
{
    private float[][] elements;
    private int rows;
    private int cols;
    public int getRows()
    {
        return rows;
    }
    public int getCols()
    {
        return cols;
    }
    public Matrix(int rows, int cols)
    {
        this.rows = rows;
        this.cols = cols;
        elements = new float[rows][cols];
    }
    public void setElement(int row, int col, float value)
    {
        elements[row][col] = value;
    }
    public float getElement(int row, int col)
    {
        return elements[row][col];
    }
    ...
    ...
```

Figure 4

Answer the following questions about the code:

i. What does the line `package ie.dit` indicate about the location of this file in the file system?

(2 marks)

ii. What is the significance of the methods `getRows()` and `getCols()`? Are they useful?

(3 marks)

iii. It is considered good practice to include a default constructor for classes. Should this class have a default constructor? Justify your answer

(3 marks)

(c) Making use of the `Matrix` class given in Figure 4 write the implementation for the method:

```
public static float LevenshteinDistance(String needle, String
haystack);
```

The purpose of this method is to return the *Levenshtein Distance* between the two `String` parameters.

(10 marks)

Question 5

(a) What is the purpose of the following classes from the Unity game engine

i. *GameObject*
ii. *GameComponent*
iii. *Transform*
iv. *Vector3*
v. *Quaternion*

(10 marks)

(b) Figure 5 shows an extract from a Unity script. Explain each line of code in the method `OnCollisionEnter`. What will happen to any GameObject the script is attached to?

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AITank : MonoBehaviour {
    private List<GameObject> allMyParts = new List<GameObject>();

    private Vector3 RandomVelocity()
    {
        return new Vector3(Random.Range(-5, 5), Random.Range(5, 10), Random.Range(-5,
5));
    }

    void OnCollisionEnter(Collision c)
    {
        if (c.gameObject.tag == "Bullet")
        {
            foreach (GameObject part in allMyParts)
            {
                Rigidbody rb = part.GetComponent<Rigidbody>();
                if (rb == null)
                {
                    rb = part.gameObject.AddComponent<Rigidbody>();
                }
                rb.isKinematic = false;
                rb.useGravity = true;
                rb.transform.parent = null;
                rb.velocity = RandomVelocity();
            }
        }
    }

    void Start () {
        allMyParts.Add(this.gameObject);
        for (int i = 0; i < transform.childCount; i++)
        {
            allMyParts.Add(transform.GetChild(i).gameObject); ;
        }

}
```

Figure 5

(15 marks)

Question 6

(a) Define the term *embedded database engine*?

(2 marks)

(b) Figure 6 shows a subset of the data stored in the *tunes* table from an sqlite database file stored in a file called *tunes.sqlite*.

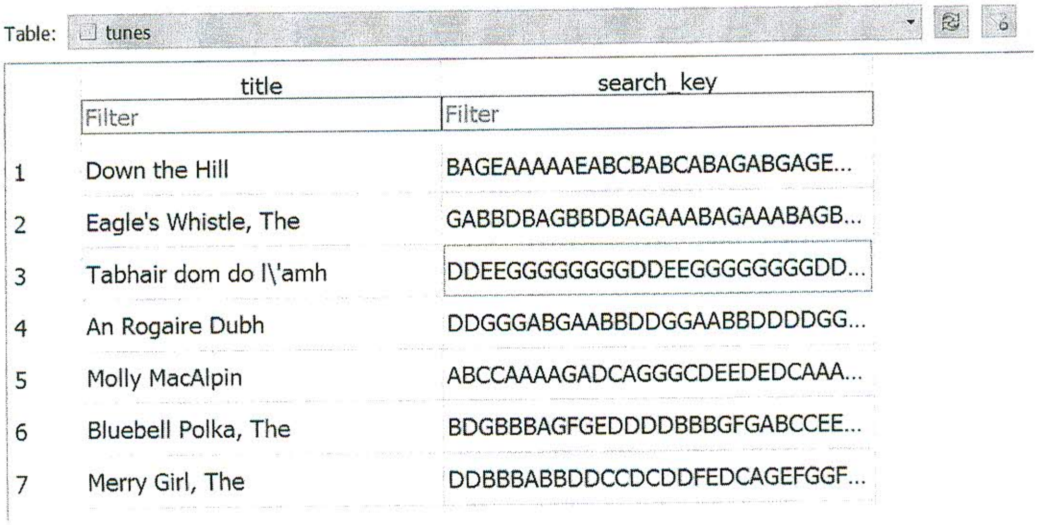

Table: tunes

	title	search_key
	Filter	Filter
1	Down the Hill	BAGEAAAAAEABCBABCABAGABGAGE...
2	Eagle's Whistle, The	GABBDBAGBBDBAGAAABAGAAABAGB...
3	Tabhair dom do l\'amh	DDEEGGGGGGGGDDEEGGGGGGGGDD...
4	An Rogaire Dubh	DDGGGABGAABBDDGGAABBDDDDGG...
5	Molly MacAlpin	ABCCAAAAGADCAGGGCDEEDEDCAAA...
6	Bluebell Polka, The	BDGBBBAGFGEDDDDBBBGFGABCCEE...
7	Merry Girl, The	DDBBBABBDDCCDCDDFEDCAGEFGGF...

Figure 6

Write code for a Java class called `Tune` that encapsulates one row of data from the file. In your solution include:

(i) Private fields to encapsulate the columns from the table.

(2 marks)

(ii) A `toString` method.

(4 marks)

(iii) A setter for the `search_key` column.

(2 marks)

(iv) A constructor that takes an instance of a ResultSet object as a parameter.

(4 marks)

(c) Write Java code to connect to the database and populate an ArrayList of Tune objects with the results of a select query that selects all the rows from the database given in Figure 6.

(11 marks)